

**PENGEMBANGAN PERANGKAT LUNAK LAYANAN ONLINE
KEBUN RAYA BOGOR BERBASIS WEB**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Elke Cahya Putri
NIM: 145150201111022



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN PERANGKAT LUNAK LAYANAN ONLINE KEBUN RAYA BOGOR
BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Elke Cahya Putri
NIM: 145150201111022

Skrripsi ini telah diuji dan dinyatakan lulus pada
18 Juli 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Bayu Priyambadha, S.Kom, M.Kom
NIP: 19820909 200812 1 004

Fajar Pradana, S.ST., M.Eng.
NIP: 19871121 201504 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Juli 2018



Elke Cahya Putri

NIM: 145150201111022

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Perangkat Lunak Layanan Online Kebun Raya Bogor” ini dapat terselesaikan. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua atas doa dan dukungan yang diberikan kepada penulis selama proses studi yang ditempuh penulis selama ini.
2. Bapak Bayu Priyambadha, S.Kom, M.Kom dan Bapak Fajar Pradana, S.ST, M.Eng selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Teknik Informatika,
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika,
5. Saudara Nova Aminarto yang telah mengizinkan penelitiannya berjudul “Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode *Local Binary Pattern*” untuk penulis kembangkan pada sistem Layanan Online Kebun Raya Bogor.
6. Saudara Richa Amalia dan khususnya untuk teman-teman keminatan Rekayasa Perangkat Lunak yang sudah bersedia meluangkan waktu untuk berdiskusi dan memberikan dukungan dalam proses pengerjaan skripsi ini.
7. Teman-teman teknik informatika 2014 atas bantuan, motivasi dan memberikan waktu untuk berdiskusi selama ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya

Malang, 18 Juli 2018

Penulis

elkecahya@student.ub.ac.id

ABSTRAK

Elke Cahya Putri, Pengembangan Perangkat Lunak Layanan Online Kebun Raya Bogor.

Pembimbing : Bayu Priyambadha, S.Kom, M.Kom dan Fajar Pradana, S.ST, M.Eng

Pusat Konservasi Tumbuhan Kebun Raya-LIPI (PKT Kebun Raya-LIPI) merupakan salah satu organisasi pemerintah yang berfokus pada bidang konservasi tumbuhan. Kebun raya Bogor merupakan salah satu kebun raya di Indonesia yang dikelola langsung oleh pihak PKT Kebun Raya-LIPI. Kebun raya Bogor memiliki luas lahan 87 hektar yang dalamnya terdapat beragam jenis tumbuhan, objek wisata dan berbagai macam fasilitas untuk pengunjung yang disediakan oleh pihak pengelola. Dengan lahan yang dimiliki sangatlah luas, dalam hal pengelolaan dan penyedia layanan, maupun dalam hal penyediaan informasi terdapat beberapa permasalahan yang muncul. Permasalahan pertama yang terjadi adalah adanya pengunjung yang merasa kesulitan menentukan *route* menuju ke salah satu objek wisata di kebun raya Bogor. Permasalahan kedua yang terjadi adalah pengunjung tidak mengetahui keanekaragaman objek wisata yang disediakan pengelola di kebun raya Bogor. Permasalahan ketiga yaitu dalam bidang pelayanan dan pengelolaan maupun penyediaan informasi adalah penanganan pemberian masukan dari pengunjung yang masih dilakukan secara manual, dengan mendatangi bagian Visitor Center atau menuliskan komentar pada laman fanpage facebook kebun raya Bogor. Untuk menangani permasalahan tersebut dilakukan pengembangan sistem yang bertujuan untuk mempermudah pengunjung mendapatkan informasi daftar objek wisata kebun raya Bogor, informasi pemetaan lokasi objek wisata di kebun raya Bogor, dan memberikan masukan atau testimoni kepada pihak pengelola. Pada fitur pemberian testimoni untuk pihak pengelola, akan dilakukan pendeteksian ekspresi mikro dari foto unggahan pengunjung dengan menggunakan algoritme *Local Binary Pattern*. Hasil dari pengujian unit, pengujian integrasi dan pengujian validasi menghasilkan nilai 100% valid. Pada pengujian kompatibilitas di dapatkan hasil bahwa sistem dapat di berjalan pada 8 jenis perambah.

Kata kunci: rekayasa perangkat lunak, *Local Binary Pattern*, kebun raya Bogor

ABSTRACT

Plant Conservation Botanic Gardens-LIPI is one of the government organizations that focuses on the field of plant conservation. Bogor Botanical Gardens is one of the botanical gardens in Indonesia which is managed directly by the Plant Conservation Botanic Gardens-LIPI. Bogor Botanical Gardens has 87 hectares in large. There are various plants, attractions and various facilities for visitors provided by the manager. Because of its large, it has many problems in managing the botanic garden, such as information providing, services, and attraction management. The first problem is there are few visitors who find it difficult to determine the route to one of the attractions in Bogor Botanical Gardens. The second problem is the visitors do not know various attractions provided by the manager at Bogor Botanical Gardens. The third problem is in the field of service and management and provision of information, it is the testimonies from visitors are handled manually, by visiting the Visitor Center or write a comment on the Bogor Botanical Garden's facebook fanpage. In order to solve those problems, the author develops a system to facilitate the visitors to get information about Bogor botanical garden attraction objects, mapping information on the location of attraction object in Bogor botanical gardens, and to provide an easier way to give the testimony to the manager. In the testimonial feature, the system detects micro expression of the uploaded photo of the visitor using the Local Binary Pattern algorithm. The results from unit testing, integration testing and validation testing are 100%. In compatibility testing, the result shows that the system can run on 8 types of browsers.

Keywords: *software engineering, Local Binary Pattern, Bogor Botanical Gardens*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Kajian Pustaka	6
2.2 Rekayasa Perangkat Lunak	7
2.3 Pengembangan Perangkat Lunak	8
2.3.1 Model <i>Waterfall</i>	8
2.3.2 Pendekatan Berorientasi Objek.....	13
2.3.3 Pemodelan Berorientasi Objek.....	13
2.4 Ekspresi Mikro.....	14
2.5 <i>Local Binary Pattern</i> (LBP)	14
2.6 Teknologi Pengembangan Sistem	15
2.6.1 Codeigniter	15
2.6.2 OpenCV	16
2.6.3 SortSite	17
2.6.4 Google Maps API	18

BAB 3 METODOLOGI	19
3.1 Studi Literatur	19
3.2 Rekayasa Kebutuhan Sistem	19
3.3 Perancangan Sistem	20
3.4 Implementasi Sistem	20
3.5 Pengujian Sistem	21
3.6 Kesimpulan dan Saran	22
BAB 4 REKAYASA KEBUTUHAN	23
4.1 Gambaran Umum Sistem	23
4.2 Identifikasi Aktor	23
4.3 Daftar Kebutuhan Fungsional	24
4.3.2 Spesifikasi Kebutuhan	26
4.4 Analisis Data	27
4.5 Daftar Kebutuhan Non-Fungsional	27
4.6 Pemodelan Kebutuhan	28
4.6.1 Use Case Diagram	28
4.6.2 Use Case Scenario	29
BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM	37
5.1 Perancangan Sistem	37
5.1.1 Perancangan Arsitektur	37
5.1.2 Perancangan Komponen	40
5.1.3 Perancangan Data	43
5.1.4 Percangan Antarmuka	44
5.2 Implementasi Sistem	53
5.2.1 Spesifikasi Sistem	53
5.2.2 Implementasi Kode Program	54
5.2.3 Implementasi Data	58
5.2.4 Implementasi Antarmuka	58
BAB 6 PENGUJIAN SISTEM	63
6.1 Pengujian Unit	63
6.1.1 Pengujian Unit <i>Method</i> Simpan	63
6.1.2 Pengujian Unit <i>Method</i> Tambah Lokasi	65

6.1.3 Pengujian Unit <i>Method</i> Edit Lokasi.....	68
6.2 Pengujian Integrasi.....	70
6.3 Pengujian Validasi	72
6.4 Pengujian Kompatibilitas	79
BAB 7 PENUTUP	81
7.1 Kesimpulan	81
7.2 Saran.....	82
DAFTAR PUSTAKA	83



DAFTAR TABEL

Tabel 2.1 Kajian pustaka	6
Tabel 2.2 Source Code Menampilkan Peta	18
Tabel 4.1 Identifikasi Aktor	23
Tabel 4.2 Daftar Kebutuhan Fungsional.....	24
Tabel 4.3 Daftar Kebutuhan Non-Fungsional.....	28
Tabel 4.4 <i>Use Case Scenario Login</i>	29
Tabel 4.5 <i>Use Case Scenario</i> Melihat Testimoni.....	29
Tabel 4.6 <i>Use Case Scenario</i> Melihat Peta	30
Tabel 4.7 <i>Use Case Scenario</i> Melihat Daftar Objek Wisata Terfavorit	30
Tabel 4.8 <i>Use Case Scenario</i> Mencari Lokasi Objek Wisata	31
Tabel 4.9 <i>Use Case Scenario Logout</i>	31
Tabel 4.10 <i>Use Case Scenario</i> Menyimpan Testimoni	32
Tabel 4.11 <i>Use Case Scenario Login Admin</i>	32
Tabel 4.12 <i>Use Case Scenario Logout Admin</i>	33
Tabel 4.13 <i>Use Case Scenario</i> Menambah Objek Wisata	33
Tabel 4.14 <i>Use Case Scenario</i> Mengubah Detail Informasi Objek Wisata.....	34
Tabel 4.15 <i>Use Case Scenario</i> Menghapus Objek Wisata	35
Tabel 4.16 <i>Use Case Scenario</i> Melihat Daftar Objek Wisata.....	35
Tabel 4.17 <i>Use Case Scenario</i> Melihat Grafik Kepuasan Pengunjung	36
Tabel 5.1 Perancangan Komponen Method <i>Simpan</i>	41
Tabel 5.2 Perancangan Komponen Method Tambah Lokasi	41
Tabel 5.3 Perancangan Komponen Method Edit Lokasi.....	43
Tabel 5.4 Perubahan Implementasi Data.....	43
Tabel 5.5 Perancangan Antarmuka Master Lokasi	44
Tabel 5.6 Perancangan Antarmuka Lihat Daftar Objek Wisata.....	46
Tabel 5.7 Perancangan Antarmuka Tambah Objek Wisata	47
Tabel 5.8 Perancangan Antarmuka Testimoni (<i>no login</i>)	49
Tabel 5.9 Perancangan Antarmuka Testimoni (<i>login</i>)	50
Tabel 5.10 Perancangan Antarmuka Jelajah	51
Tabel 5.11 Perancangan Antarmuka Kelola Objek Wisata	52

Tabel 5.12 Spesifikasi Perangkat Keras	53
Tabel 5.13 Spesifikasi Perangkat Lunak	54
Tabel 5.14 Sistem Operasi	54
Tabel 5.15 Implementasi Kode Program <i>Method</i> Simpan	54
Tabel 5.16 Implementasi Kode Program <i>Method</i> Tambah Lokasi	55
Tabel 5.17 Implementasi Kode Program <i>Method</i> Edit Lokasi	56
Tabel 5.18 Implementasi Data Layanan Online Kebun Raya Bogor	58
Tabel 6.1 <i>Pseudocode</i> Pengujian Unit Method Simpan	63
Tabel 6.2 Hasil Pengujian Unit Mim stillethod Simpan	64
Tabel 6.3 <i>Pseudocode</i> Pengujian Unit <i>Method</i> Tambah Lokasi	65
Tabel 6.4 Hasil Pengujian Unit Method Tambah Lokasi	67
Tabel 6.5 <i>Pseudocode</i> Pengujian Unit <i>Method</i> Edit Lokasi	68
Tabel 6.6 Langkah Uji Pengujian Integrasi	70
Tabel 6.7 Source Code Method Simpan Kondisi Benar	71
Tabel 6.8 <i>Source Code Method</i> Stub_save	71
Tabel 6.9 Source Code Method Simpan Kondisi Salah	72
Tabel 6.10 Pengujian Validasi <i>Login</i>	72
Tabel 6.11 Pengujian Validasi Login Alternatif 1	73
Tabel 6.12 Pengujian Validasi Melihat Testimoni	73
Tabel 6.13 Pengujian Validasi Melihat Peta	73
Tabel 6.14 Pengujian Validasi Melihat Daftar Objek Wisata Terfavorit	74
Tabel 6.15 Pengujian Validasi Mencari Lokasi Objek Wisata	74
Tabel 6.16 Pengujian Validasi <i>Logout</i>	75
Tabel 6.17 Pengujian Validasi Menyimpan Testimoni	75
Tabel 6.18 Pengujian Validasi Menyimpan Testimoni Alternatif 1	75
Tabel 6.19 Pengujian Validasi <i>Login</i> Admin	76
Tabel 6.20 Pengujian Validasi Login Admin Alternatif 1	76
Tabel 6.21 Pengujian Validasi <i>Logout</i> Admin	76
Tabel 6.22 Pengujian Validasi Menambah Objek Wisata	77
Tabel 6.23 Pengujian Validasi Menambah Objek Wisata Alternatif 1	77
Tabel 6.24 Pengujian Validasi Menambah Objek Wisata Alternatif 2	78
Tabel 6.25 Pengujian Validasi Mengubah Detail Informasi Objek Wisata	78

Tabel 6.26 Pengujian Validasi Menghapus Objek Wisata	78
Tabel 6.27 Pengujian Validasi Melihat Daftar Objek Wisata	79
Tabel 6.28 Pengujian Validasi Melihat Grafik Kepuasan Pengunjung	79
Tabel 6.29 Pengujian Kompatibilitas	80



DAFTAR GAMBAR

Gambar 2.1 Model Waterfall	8
Gambar 2.2 Complexity Number	11
Gambar 2.3 Jarak dan Banyak Piksel Tetangga yang Terpilih (a)R=1; P=8, (b)R=2; P=16, (c)R=2; P=8.....	14
Gambar 2.4 Proses Kalkulasi Piksel LBP.....	15
Gambar 2.5 Hubungan Komponen MVC	16
Gambar 2.6 Struktur OpenCV.....	17
Gambar 3.1 Diagram Alir Penelitian	19
Gambar 4.1 ERD Layanan Online Kebun Raya Bogor	27
Gambar 4.2 <i>Use Case Diagram</i> Layanan Online Kebun Raya Bogor	28
Gambar 5.1 Sequence Diagram Menyimpan Testimoni.....	37
Gambar 5.2 Sequence Diagram Mengubah Detail Informasi Objek Wisata.....	38
Gambar 5.3 <i>Sequence Diagram</i> Mengubah Detail Informasi Objek Wisata.....	39
Gambar 5.4 <i>Class Diagram</i> Layanan Online Kebun Raya Bogor	40
Gambar 5.5 PDM Layanan Online Kebun Raya Bogor	44
Gambar 5.6 Perancangan Antarmuka Master Lokasi	44
Gambar 5.7 Perancangan Antarmuka Lihat Daftar Objek Wisata.....	45
Gambar 5.8 Perancangan Antarmuka Tambah Objek Wisata	47
Gambar 5.9 Perancangan Antarmuka Testimoni (<i>no login</i>)	48
Gambar 5.10 Perancangan Antarmuka Testimoni (<i>login</i>)	49
Gambar 5.11 Perancangan Antarmuka Jelajah	51
Gambar 5.12 Perancangan Antarmuka Kelola Objek Wisata	52
Gambar 5.13 Implementasi Antarmuka Halaman Master Lokasi	58
Gambar 5.14 Implementasi Antarmuka Halaman Tambah Objek Wisata	59
Gambar 5.15 Implementasi Antarmuka Halaman Admin Lihat Daftar Objek Wisata	59
Gambar 5.16 Implementasi Antarmuka Halaman Testimoni (<i>no login</i>).....	60
Gambar 5.17 Implementasi Antarmuka Halaman Testimoni (<i>login</i>)	60
Gambar 5.18 Implementasi Antarmuka Halaman Jelajah	61
Gambar 5.19 Implementasi Antarmuka Halaman Pengelolaan Objek Wisata	62

Gambar 6.1 <i>Flow Graph Method</i> Simpan	64
Gambar 6.2 <i>Flow Graph Method</i> Tambah Lokasi.....	66
Gambar 6.3 <i>Flow Graph Method</i> Edit Lokasi	69
Gambar 6.4 Diagram Hierarki Pengujian Integrasi <i>Method</i> Simpan	70
Gambar 6.5 Hasil Pengujian Method stub_save Kondisi Benar	71
Gambar 6.6 Hasil Pengujian Method stub_save Kondisi Salah.....	72
Gambar 6.7 Hasil Pengujian Kompatibilitas	80



DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA PENGUNJUNG KEBUN RAYA BOGOR	85
LAMPIRAN B KUESIONER PENGUNJUNG KEBUN RAYA BOGOR.....	87
LAMPIRAN C DAFTAR KEBUTUHAN PERANGKAT LUNAK LAYANAN ONLINE KEBUN RAYA BOGOR	92
LAMPIRAN D PEMODELAN KEBUTUHAN PERANGKAT LUNAK LAYANAN ONLINE KEBUN RAYA BOGOR	94
LAMPIRAN E PERANCANGAN ARSITEKTUR PERANGKAT LUNAK LAYANAN ONLINE KEBUN RAYA BOGOR	95
LAMPIRAN F PERANCANGAN DATA PERANGKAT LUNAK LAYANAN ONLINE KEBUN RAYA BOGOR	102



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pusat Konservasi Tumbuhan Kebun Raya-LIPI (PKT Kebun Raya-LIPI) merupakan salah satu organisasi pemerintah yang berfokus pada bidang konservasi tumbuhan. Kebun raya Bogor merupakan salah satu kebun raya di Indonesia yang dikelola langsung oleh pihak PKT Kebun Raya-LIPI. Kebun raya Bogor memiliki fungsi utama yaitu sebagai pusat konservasi tumbuhan, selain itu dapat difungsikan juga sebagai pusat penelitian tumbuhan, tempat edukasi dan tempat wisata (Perpres, 2011). Lokasi kebun raya Bogor yang berada di Jalan Ir. Haji Djuanda No.13, Kota Bogor. Kebun raya Bogor memiliki luas lahan mencapai 87 hektar, yang di dalamnya terdapat berbagai fasilitas dan objek wisata yang dikelola langsung oleh pihak PKT Kebun Raya-LIPI.

Dalam hal pengelolaan objek wisata di kebun raya Bogor terdapat beberapa masalah yang terjadi, yaitu seperti pemetaan lokasi objek wisata kebun raya Bogor yang hanya dapat dilihat pada peta kebun raya Bogor, baik peta yang terdapat pada setiap kantor pihak pengelola maupun peta yang ada di sekitar tempat pintu masuk kebun raya Bogor. Selain itu, masalah yang sering terjadi juga terdapat beberapa tumbuhan langka yang ditanam tidak pada lokasi yang tetap, sehingga jika terjadi perpindahan lokasi memerlukan waktu yang banyak untuk melakukan update data terbaru untuk di lakukan mencetak ulang. Untuk meningkatkan pelayanan yang diberikan untuk pengunjung kebun raya Bogor, pihak pengelola merasa kesulitan mendapatkan data akan kepuasan pengunjung yang hanya berdasarkan testimoni yang diberikan pada laman *fanpage* facebook kebun raya Bogor maupun pada bagian Visitor Center. Data testimoni yang diberikan pengunjung kebun raya Bogor selama ini banyak hanya berupa kata-kata saja, hal itu dirasa kurang *valid* dalam merepresentasikan bagaimana tingkat kepuasan pengunjung.

Selain itu dalam hal penyedia informasi, pihak pengelola kebun raya Bogor telah menyediakan salah satu *website* yang bernama Layanan Online Kebun Raya Bogor untuk memberikan informasi terkait dengan fasilitas-fasilitas yang ada di kebun raya Bogor. Salah satu fasilitas yang disediakan yaitu aneka ragam objek wisata yang dapat dikunjungi oleh pengunjung kebun raya Bogor. Pada *website* tersebut pengunjung juga dapat melakukan pemesanan terkait dengan fasilitas yang ada di kebun raya Bogor. Namun, berdasarkan dari hasil wawancara dengan salah satu pengunjung kebun raya Bogor di dapatkan beberapa permasalahan yang terkait dengan penyediaan informasi yang kurang untuk pengunjung. Permasalahan yang terjadi yaitu, dengan adanya *website* tersebut pengunjung masih merasakan kesulitan dalam mendapatkan aneka ragam informasi objek wisata yang ada di kebun raya Bogor. Pada halaman *website* tersebut, hanya memberikan informasi terkait dengan fasilitas yang hanya bisa dilakukan pemesanan lah yang tercantum, seperti fasilitas gedung konservasi, fasilitas sewa lapangan dan sebagainya. Kenyataanya, pengunjung ingin mendapatkan informasi

lebih terkait dengan fasilitas-fasilitas yang disediakan oleh pihak pengelola tapi tidak tersedia.

Permasalahan yang kedua yaitu pengunjung merasakan kesulitan dalam melakukan pencarian lokasi objek wisata. Pada halaman *website* tersebut hanya memberikan informasi bagaimana cara pemesanan dan melakukan pembayaran saja tanpa memberitahukan lokasi dimana objek tersebut berada. Permasalahan yang ketiga yaitu, banyaknya pengunjung yang merasa tidak mengetahui keanekaragaman objek wisata yang ada di kebun raya Bogor, hal itu dibuktikan dengan hasil dari penyebaran kuesioner terkait dengan seberapa familiar kah pengunjung terhadap ragam objek wisata yang ada di kebun raya Bogor pada Lampiran B.

Menurut INPRES 2003 dalam bidang peningkatan strategi E-Government, dalam mengembangkan sistem pelayanan harus andal dan terpercaya, serta terjangkau oleh masyarakat luas. Berdasarkan Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 38 Tahun 2008 tentang Pengelolaan Teknologi Informasi dan Komunikasi di Lingkungan Departemen Pendidikan Nasional pasal 1 butir 4, TIK Departemen adalah teknologi informasi dan komunikasi berbasis elektronika yang digunakan oleh Departemen untuk melakukan pengambilan, pengumpulan, pengolahan, penyimpanan, penyebaran, dan penyajian informasi. Dalam hal penyebaran dan penyajian informasi, perlu untuk diketahui tingkat efektifitas layanan informasi situs web resmi suatu instansi pemerintahan. Untuk itu dibutuhkan penambahan layanan masukkan atau testimoni untuk menjangkau setiap testimoni yang diberikan pengunjung kebun raya Bogor yang berada pada halaman situs web resmi Layanan Online Kebun Raya Bogor. Selain itu, penelitian yang dilakukan oleh TripAdvisor (2014) yaitu situs perjalanan terbesar di dunia ini mengumumkan hasil dari penelitian independen yang menekankan betapa pentingnya review wisatawan online dalam proses pencarian dan pemesanan. Menurut penelitian tersebut lebih dari 80% mengatakan bahwa review dari suatu situs membantu mereka percaya diri dalam mengambil keputusan wisata dan membantu mereka mendapatkan perjalanan yang lebih baik, sehingga jika halaman *website* Layanan Online Kebun Raya Bogor diberikan penambahan fitur terkait dengan testimoni pengunjung, hal ini dapat memberikan dampak positif terhadap pemasukkan yang di dapat pihak pengelola kebun raya Bogor.

Berdasarkan dari permasalahan yang di dapatkan, bahwa masih banyak permasalahan dalam hal pengelolaan dan penyedia informasi yang belum terselesaikan dengan adanya *website* Layanan Online Kebun Raya Bogor, maka solusi yang diberikan yaitu akan dilakukannya pengembangan lanjut dalam penelitian yang berjudul "Pengembangan Perangkat Lunak Layanan Online Kebun Raya Bogor Berbasis Web". Pengembangan lanjut ini menambahkan beberapa fitur yang digunakan untuk mempermudah pihak pengelola dalam melakukan pengelolaan lokasi objek wisata, melakukan penambahan atau perubahan detail informasi objek wisata baik lokasi maupun nama dan gambar pendukung objek wisata tersebut, mendapatkan data *valid* terkait dengan tingkat kepuasan

pengunjung untuk mendapatkan ekspresi natural para pengunjung pada fitur testimoni yang diberikan pengunjung dan dapat mempermudah para pengunjung untuk mendapatkan informasi terkait dengan lokasi dan ragam objek wisata yang ada di kebun raya Bogor. Pengembangan lanjut pada website Layanan Online Kebun Raya Bogor yang dipilih masih berbasis web, dikarenakan lebih mempermudah pengunjung dalam memperoleh informasi tanpa harus ketergantungan pada salah satu *platform* khusus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat diambil rumusan masalah sebagai berikut :

1. Apa saja kebutuhan untuk pengembangan perangkat lunak Layanan Online Kebun Raya Bogor?
2. Bagaimana hasil perancangan dan implementasi perangkat lunak Layanan Online Kebun Raya Bogor menyelesaikan masalah yang ada?
3. Bagaimana hasil dari pengujian memenuhi kebutuhan perancangan perangkat lunak yang dikembangkan?

1.3 Tujuan

Tujuan yang akan dicapai dari penelitian pengembangan perangkat lunak Layanan Online Kebun Raya Bogor berbasis web yaitu :

1. Menganalisis kebutuhan sistem Layanan Online Kebun Raya Bogor.
2. Merancang sistem dan mengimplementasikan rancangan yang telah dibuat kedalam kode program berbasis web.
3. Menguji sistem Layanan Online Kebun Raya Bogor telah sesuai dengan kebutuhan.

1.4 Manfaat

Manfaat dari adanya penelitian pengembangan perangkat lunak Layanan Online Kebun Raya Bogor yaitu :

1. Dapat menyediakan perangkat lunak yang dapat digunakan untuk proses pemasaran fasilitas dan destinasi objek wisata yang tersedia di kebun raya Bogor.
2. Dapat mempermudah melakukan evaluasi pelayanan pihak Pusat Konservasi Tumbuhan Kebun Raya-LIPI.
3. Dapat mempermudah pengunjung mendapatkan informasi terkait dengan kebun raya Bogor.

1.5 Batasan Masalah

Untuk menghindari terjadinya penyimpangan dalam penelitian ini, maka dirumuskan beberapa batasan-batasan, yaitu :

1. Fokus pengenalan ekspresi mikro pada bagian wajah.

2. Pengenalan ekspresi dilakukan pada gambar diam dan hanya pada satu wajah yang terdapat pada gambar.
3. Sistem dibangun dengan menggunakan bahasa pemrograman PHP, Javascript, HTML, CSS dan *framework* PHP Codeigniter.
4. Sistem menggunakan basis data MYSQL.

1.6 Sistematika Pembahasan

Sistematika penulisan pada penelitian ini berfungsi untuk agar lebih mudah memahami alur dalam penelitian dan memberikan informasi terkait dengan garis besar yang ada, yaitu sebagai berikut :

BAB 1 : PENDAHULUAN

Pada bab pendahuluan akan menjelaskan tentang latar belakang dari pengembangan perangkat lunak Layanan Online Kebun Raya Bogor, akan menjelaskan tentang rumusan masalah dari penelitian, batasan masalah penelitian, tujuan penelitian, manfaat dari penelitian yang dilakukan dan sistematika dari penulisan.

BAB 2 : LANDASAN KEPUSTAKAAN

Pada bab landasan kepastakaan berisi mengenai bagaimana penjelasan terkait dengan teori-teori yang akan digunakan sebagai bahan referensi dalam pengembangan perangkat lunak seperti teori terkait dengan *Local Binary Pattern* (LBP), Rekayasa Perangkat Lunak, model dalam pengembangan perangkat lunak, teori ekspresi mikro, opencv dan lain sebagainya yang digunakan pada penelitian ini.

BAB 3 : METODOLOGI

Pada bab metodologi berisi terkait dengan urutan atau susunan penelitian, dimulai dari melakukan analisis kebutuhan atau rekayasa kebutuhan, melakukan perancangan dan mengembangkan perangkat lunak Layanan Online Kebun Raya Bogor.

BAB 4 : REKAYASA KEBUTUHAN

Pada bab rekayasa kebutuhan akan membahas tentang analisis kebutuhan dan pemodelan kebutuhan yang akan diterapkan dalam pengembangan sistem Layanan Online Kebun Raya Bogor. Pada bab rekayasa kebutuhan ini berisi seperti analisis kebutuhan sistem, analisis data, daftar kebutuhan sistem dan pemodelan kebutuhan sistem.

BAB 5 : PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab perancangan dan implementasi sistem akan membahas tentang perancangan dan implementasi sistem Layanan Online Kebun Raya Bogor. Pada bab ini terdapat beberapa perancangan yang akan dilakukan diantaranya yaitu perancangan arsitektur, perancangan komponen, perancangan data perancangan antarmuka. Pada tahap implementasi akan menjelaskan terkait dengan spesifikasi sistem yang digunakan pada saat pengembangan sistem,

implementasi kode program, implementasi data, dan implementasi antarmuka sistem.

BAB 6 : PENGUJIAN SISTEM

Pada bab pengujian sistem akan membahas terkait dengan proses pengujian yang akan dilakukan pada sistem yang dikembangkan. Pengujian yang akan dibahas yaitu tentang pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian kompatibilitas.

BAB 7 : PENUTUP

Pada bab penutup akan menjelaskan terkait dengan kesimpulan dari hasil penelitian yang dilakukan dan saran pengembangan untuk sistem Layanan Online Kebun Raya Bogor dikemudian hari.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada kajian pustaka akan membahas tentang penelitian-penelitian sebelumnya yang dijadikan referensi dalam penelitian ini dapat dilihat pada Tabel 2.1 yaitu sebagai berikut :

Tabel 2.1 Kajian pustaka

No	Judul Penelitian	Objek Penelitian	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	Facial Micro-Expression Recognition using Spatiotemporal <i>Local Binary Pattern</i> with Integral Projection (Xiaohua Huang, et al., 2015).	Pengenalan emosi berdasarkan ekspresi mikro wajah menggunakan <i>Local Binary Pattern</i> dengan <i>Integral Projection</i> .	Pengenalan emosi berdasarkan ekspresi mikro wajah menggunakan <i>Local Binary Pattern</i> dengan <i>Integral Projection</i> .	Pengenalan emosi berdasarkan ekspresi mikro wajah menggunakan <i>Local Binary Pattern</i> .
2.	CASME II: An Improved Spontaneous Micro-Expression Basis data and the Baseline Evaluation (Wen-Jing Yan, et al., 2014).	Membangun basis data untuk penentuan ekspresi mikro yang di beri nama CASME II.	Membangun basis data untuk penentuan ekspresi mikro yang di beri nama CASME II.	Menggunakan data dari CASME II untuk melakukan pendeteksian ekspresi mikro.
3.	Penerapan Teknik Pengenalan Wajah Berbasis Fitur Local Binary Pattern pada Robot Pengantar Makanan	Pengenalan wajah menggunakan fitur <i>Haar-Cascade</i> untuk deteksi wajah dan menggunakan	Pengenalan wajah menggunakan fitur <i>Haar-Cascade</i> untuk deteksi wajah dan menggunakan	Menggunakan fitur <i>Haar-Cascade</i> pada OpenCV yang digunakan untuk pendeteksian wajah.

	(Turiyanto, et al., 2014).	metode <i>Local Binary Pattern</i> .	metode <i>Local Binary Pattern</i> .	
4.	Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode <i>Local Binary Pattern</i> (Aminarto, N., Sari, Y., & Wihandika, R).	Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode <i>Local Binary Pattern</i> .	Pengenalan ekspresi mikro menggunakan metode <i>Local Binary Pattern</i> dengan memanfaatkan library OpenCV untuk pendeteksian wajah.	Pengenalan ekspresi mikro menggunakan metode <i>Local Binary Pattern</i> dengan memanfaatkan library OpenCV untuk pendeteksian wajah.
5.	Pengembangan sistem informasi tanaman di kebun raya Bogor berbasis android (M. Julian Pagurawan, 2015).	Pengembangan sistem informasi tanaman di kebun raya Bogor berbasis android.	Membuat sistem informasi tanaman yang ada di kebun raya Bogor berbasis android dengan memanfaatkan <i>Google Maps</i> dan data koordinat lokasi kebun raya Bogor.	Membuat sistem informasi yang terkait dengan pelayanan dan fasilitas yang ada di kebun raya Bogor dan memanfaatkan <i>Google Maps</i> untuk menentukan lokasi objek wisata kebun raya Bogor.

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan disiplin ilmu yang berkaitan dengan keseluruhan aspek-aspek produksi perangkat lunak, dimulai dari tahap awal spesifikasi sistem sampai dengan tahap pemeliharaan sistem setelah sistem mulai digunakan (Sommerville, 2011). Pada definisi rekayasa perangkat lunak terdapat dua istilah kata kunci yaitu :

1. *Engineering discipline*

Perekayasa membuat suatu alat dapat yang bekerja dengan menerapkan teori, metode, dan alat bantu yang sesuai dengan permasalahannya. Namun, para perekayasa harus menemukan solusi untuk permasalahannya dengan selektif, walaupun tidak ada teori atau metode pendukung untuk menyelesaikannya. Perekayasa mengetahui bahwa mereka harus bekerja dalam batasan organisasi

dan keuangan, sehingga mereka berusaha mencari solusi untuk menyelesaikan batasan-batasannya.

2. All aspects of software production

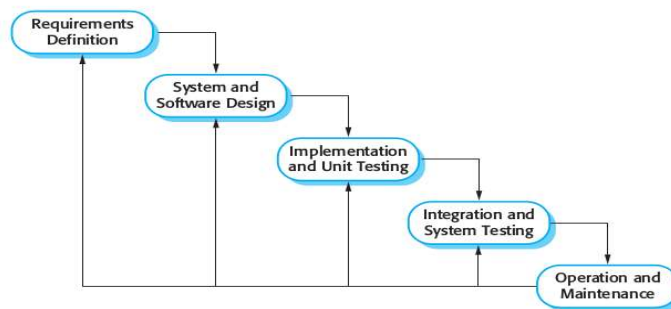
Rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak saja, akan tetapi berhubungan juga dengan kegiatan manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

2.3 Pengembangan Perangkat Lunak

Dalam proses pengembangan perangkat lunak terdapat siklus untuk membantu dalam proses pengembangan yang biasa disebut dengan *Software Development Life Cycle* (SDLC). Banyak pengembang perangkat lunak menjadikan SDLC sebagai bahan acuan dalam pengembangan perangkat lunaknya. SDLC disebut juga sebagai *Software Process Model* yang merupakan representasi sederhana dari sebuah proses perangkat lunak atau *Software Process* (Sommerville, 2011). Terdapat banyak jenis model SDLC yaitu seperti model *waterfall*, *prototyping*, *iterative and incremental development*, *spiral development*, *rapid application development*, dan *extreme programming*.

2.3.1 Model Waterfall

Model *waterfall* merupakan model pertama dalam proses pengembangan perangkat lunak yang sering disebut juga sebagai *classic life cycle*. Model waterfall ini merupakan pendekatan yang sistematis dan berurutan (sekuensial) untuk pengembangan perangkat lunak yang dimulai pada tahap spesifikasi kebutuhan sampai dengan puncaknya pada *operation and maintenance* perangkat lunak (Pressman, 2010). Model *waterfall* merupakan salah satu model SDLC yang mengambil proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi yang kemudian direpresentasikan dalam proses atau tahapan yang berbeda-beda (Sommerville, 2011). Tahapan-tahapan dalam model *waterfall* yaitu seperti *requirement definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance* yang digambarkan pada Gambar 2.1.



Gambar 2.1 Model Waterfall

Sumber : (Sommerville, 2011)

1. Requirements Definition

Requirements analysis atau *requirements definition* merupakan tahap awal dalam proses pengembangan perangkat lunak pada model *waterfall*. Tujuan dari tahap ini adalah memahami kebutuhan pengguna dan dokumen secara benar dan mengidentifikasi apa saja yang dibutuhkan oleh sistem. Output dari tahap ini adalah *Software Requirement Specification (SRS)* (Kumar, 2013).

2. System and software design

Perancangan perangkat lunak mendeskripsikan struktur dari perangkat lunak yang nantinya akan di implementasikan. Terdapat beberapa proses dalam tahap perancangan pada pengembangan perangkat lunak, yaitu :

a. Perancangan basis data

Perancangan basis data yaitu melakukan transformasi model domain informasi dari hasil analisis struktur data yang di modelkan dengan *Entity Relationship Diagram (ERD)*. Menurut Connolly dan Begg (2010), *Entity Relation Diagram (ERD)* digunakan untuk menggambarkan struktur *logical* basis data yang menyediakan cara yang sederhana dan mudah untuk memahami bagian dari komponen dalam perancangan basis data. Perancangan basis data dimodelkan dengan menggunakan *Physical Data Model (PDM)* yang akan menjadi dasar dalam implementasinya. Menurut Rosa A.S dan M.Salahuddin (2011), *Physical Data Model (PDM)* adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik beserta tipe datanya.

b. Perancangan arsitektural

Mengidentifikasi struktur keseluruhan sistem, komponen utama (sub-sistem), relasi antar komponen utama dari sistem bagaimana dan bagaimana didistribusikan.

c. Perancangan antarmuka

Menjelaskan bagaimana perangkat lunak bisa berkomunikasi didalam perangkat lunak itu sendiri maupun dengan sistem lain yang berinteraksi atau dengan manusia sebagai penggunanya.

d. Perancangan komponen

Menggambaran rincian sub-sistem dari setiap komponen perangkat lunak. Perancangan komponen harus mendefinisikan struktur data dan rincian algoritme untuk semua proses yang terjadi di dalam komponen. Rincian algoritme dituliskan dalam bentuk *pseudocode*. *Pseudocode* adalah suatu bahasa pemrograman informal dan sangat fleksibel, yang tidak dimaksudkan untuk dieksekusi pada mesin, tetapi hanya digunakan untuk mengatur pemikiran pemrogram sebelum melakukan pengkodean (Pressman, 2010).

3. Implementation and unit Testing

Pada tahap ini perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Dimana nantinya pengujian unit akan melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya.

4. Integration and system testing

Pada tahap ini masing-masing unit program diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Hal ini dilakukan untuk memastikan bahwa kebutuhan perangkat lunak semua telah terpenuhi.

Dalam tahap melakukan pengujian perangkat lunak, terdapat dua teknik pengujian yang dapat digunakan yaitu sebagai berikut (Pressman, 2010):

a. White-box testing

White-box testing adalah sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk didapatkan kasus uji. Tujuan dari *white-box testing* yang pertama adalah menjamin bahwa semua jalur yang sudah di definisikan pada jalur independen, setidaknya dilakukan eksekusi minimal sekali. Kedua, melakukan semua keputusan logis dengan kondisi benar atau salah. Ketiga, melakukan semua perulangan pada *boundaries* dan dalam batas-batas operasional dan yang terakhir melatih struktur data internal untuk memastikan validitasnya. Terdapat dua jenis teknik pengujian *white-box testing*, yaitu *basis path testing* dan *control structure testing*. Teknik *basis path testing* memungkinkan perancangan kasus uji mendapatkan ukuran kompleksitas logika dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. *Basis path testing* memastikan bahwa setidaknya setiap jalur alur logika dapat dijalankan satu kali.

Langkah-langkah dalam pembuatan *test case* pada *basis path testing* adalah sebagai berikut :

1. Pembuatan diagram alir dari perancangan prosedural atau *source code*
2. Menentukan nilai *cyclomatic complexity* dari diagram alir
3. Menentukan independent path dari diagram alir

Perhitungan *cyclomatic complexity* $V(G)$ dapat dilakukan dengan perhitungan manual dengan rumus seperti dibawah ini (Guru99, 2018) :

$$V(G) = E - N + 2$$

Keterangan :

E = Jumlah *edge*

N = Jumlah *node*

$$V(G) = P + 1$$

Keterangan :

P = Predicate Node

Dan $V(G)$ dapat dicari dengan menghitung jumlah *region* pada *flow graph*. Nilai cyclomatic complexity memiliki arti seperti yang tertuliskan pada Gambar 2.3.

Complexity Number	Meaning
1-10	Structured and well written code High Testability Cost and Effort is less
10-20	Complex Code Medium Testability Cost and effort is Medium
20-40	Very complex Code Low Testability Cost and Effort are high
>40	Not at all testable Very high Cost and Effort

Gambar 2.2 Complexity Number

Sumber: (Guru99, 2018)

Pengujian *white-box* memiliki beberapa jenis pengujian, diantaranya yaitu pengujian unit dan pengujian integrasi. Pengujian unit merupakan pengujian yang berkonsentrasi pada setiap unit (misalnya: komponen, klas, atau objek) dari perangkat lunak yang diterapkan. Pengujian integrasi, pengujian ini berfokus pada desain dan konstruksi dari arsitektur perangkat lunak (Pressman, 2010). Pada pengujian integrasi terdapat beberapa teknik pengujian yaitu *Top Down Testing*, *Bottom Up Testing*, dan *sandwich*. Pada teknik pengujian *top-down*, unit terintegrasi akan bergerak ke bawah melalui hierarki kontrol yang dimulai dari modul utamanya. *Stubs* menggantikan modul tingkat bawah. *Software test stubs* adalah program yang mensimulasikan perilaku komponen perangkat lunak (unit) yang memiliki ketergantungan, berdasarkan dari unit paling bawah. Teknik pengujian *bottom-up* adalah pengujian yang dilakukan dari unit terendah pada struktur program. Teknik pengujian *bottom-up* ini menggunakan modul *driver* untuk melakukan kontrol modul. *Software test driver* adalah program yang

mensimulasikan perilaku komponen perangkat lunak (unit) yang merupakan unit kontrol untuk teknik ini.

b. *Black-box testing*

Black-box testing merupakan sebuah pendekatan yang memiliki tujuan untuk melakukan pengujian sistem berdasarkan spesifikasi sistem tanpa harus melakukan akses atau melakukan uji terhadap kode program sistem (Sommerville, 2011). *Black-box testing* disebut juga dengan *behavioral testing* karena digunakan secara khusus pada kebutuhan fungsional sistem. *Black-box testing* dilakukan untuk mengetahui kesesuaian dari suatu fungsi tertentu dari suatu produk dengan tujuan awal dirancangnya suatu produk tersebut. Tujuan dari *black-box testing* adalah untuk menemukan *error* ketika suatu skrip dijalankan. *Black-box testing* mengkategorikan kesalahan yang ditemukan dalam beberapa kategori, antara lain:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

Black-box testing dilakukan pada saat tahap akhir pengujian, tidak seperti *white-box testing* yang dilakukan pada saat awal proses pengujian dikarenakan *black-box testing* memperhatikan dari struktur kontrol, maka pengujian ini berfokus pada domain informasi. Selain melakukan pengujian terhadap kebutuhan fungsionalitas sistem, pengujian juga dilakukan untuk menguji kebutuhan non-fungsional sistem. Pengujian kebutuhan non-fungsional biasanya mengacu pada aspek perangkat lunak yang mungkin tidak terkait dengan fungsi tertentu atau tindakan pengguna. Salah satunya yaitu pengujian kompatibilitas yang bertujuan untuk memastikan bahwa perangkat lunak yang telah dibangun mampu berjalan dengan baik pada lingkungan yang heterogen. Pada pengujian kompatibilitas ini juga melibatkan pemilihan konfigurasi lingkungan tertentu (Yoon, et al., 2008).

5. Operation and Maintenance

Pada tahap ini biasanya memerlukan waktu yang paling lama dibandingkan dengan tahapan-tahapan sebelumnya. Perangkat lunak yang telah dipasang dan dikondisikan pada kondisi penggunaan yang sebenarnya. Dalam proses pemeliharaan melibatkan proses memperbaiki *error* yang mana tidak ditemukan pada tahap awal pengembangan, meningkatkan implementasi unit sistem dan meningkatkan layanan sistem ketika ditemukan kebutuhan baru. Tahap ini tidak selalu ada dalam proses pengembangan sistem. Pada penelitian ini tidak dilakukannya tahapan *operation and maintenance*.

2.3.2 Pendekatan Berorientasi Objek

Pendekatan berorientasi object atau biasa disebut *Object Oriented* (OO) adalah salah satu pendekatan dalam pengembangan perangkat lunak. Proses pembangunan perangkat lunak menggunakan OO dimulai dari analisis kebutuhan yang disebut dengan *Object Oriented Analysis* (OOA) untuk mengidentifikasi kebutuhan-kebutuhan sistem, objek, dan kelas yang didapat dari domain permasalahan. Dari tahap OOA dilanjutkan ke tahap perancangan yang disebut *Object Oriented Design* (OOD) yang berdasar pada hasil analisis pada tahap sebelumnya. Proses pada OOD terdiri dari perancangan arsitektur, komponen, data, dan antarmuka. Proses selanjutnya adalah mengubah hasil perancangan yang sudah dilakukan menjadi bentuk kode program yang dapat dibaca mesin menggunakan bahasa pemrograman berorientasi objek, proses tersebut dinamakan *Object Oriented Programming* (OOP) yang pada akhirnya akan diuji dengan suatu proses pengujian yang dinamakan *Object Oriented Testing* (OOT), dimana dalam proses ini menguji hasil perancangan arsitektur, antarmuka, dan komponen (Pressman, 2010).

2.3.3 Pemodelan Berorientasi Objek

Pada OOA dan OOD dapat dimodelkan menggunakan diagram yang terdapat pada *Unified Modelling Language* (UML) (Pressman, 2010). UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak (Rumbaugh, 2005). Terdapat tiga diagram yang akan digunakan pada penelitian ini, yaitu (Rumbaugh, 2005):

1. Diagram use case

Diagram *use case* memberikan gambaran umum mengenai interaksi antara sistem dengan lingkungannya untuk memberikan informasi yang mudah dipahami mengenai apa saja yang terlibat dengan sistem. Diagram *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

2. Diagram sequence

Diagram *sequence* menggambarkan interaksi antar objek didalam dan diluar sistem berupa pesan yang digambarkan terhadap waktu. Diagram *sequence* merepresentasikan skenario atau rangkaian langkah-langkah yang dilakukan sebagai tanggapan dari sebuah pesan untuk menghasilkan keluaran tertentu.

3. Diagram class

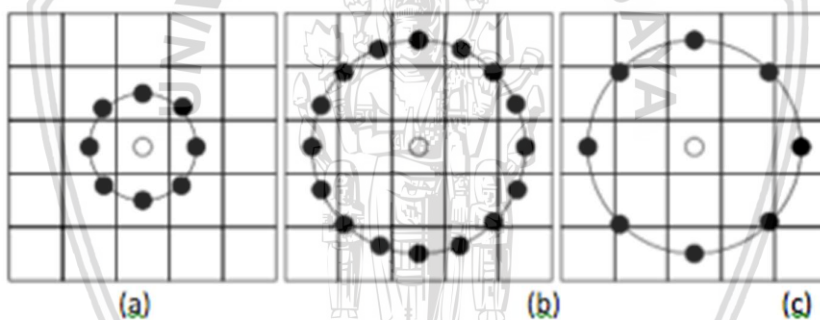
Diagram class menggambarkan struktur dan deskripsi kelas, package, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. Kelas adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan atribut suatu sistem, sekaligus menawarkan layanan untuk memanipulasi atribut tersebut dengan *method* atau fungsi.

2.4 Ekspresi Mikro

Ekspresi mikro merupakan ekspresi yang muncul secara halus dan tidak disadari oleh seseorang yang dapat berfungsi untuk mengkoordinasikan interaksi sosial melalui fungsi informatif dan memotivasi (Kelter & Kring, 1998). Selain itu ekspresi mikro yang didapatkan, dapat berfungsi untuk memberikan suatu informasi tentang emosi suasana hati seseorang (Scherer, 1986). Menurut teori Darwin tentang seleksi alam (1872), emosi bersifat universal dalam hal memiliki sejenis pola ekspresi dan fungsi pada tingkat komunikasi antar manusia, terlepas dari bias budaya. Emosi yang universal didefinisikan oleh Ekman (2003) yaitu ekspresi terkejut, bahagia, marah, sedih, takut, dan jijik.

2.5 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) pertama kali diperkenalkan oleh Ojala et al, didefinisikan sebagai ukuran tekstur *gray-scale* yang invarian, disebut invarian karena hampir tidak dipengaruhi oleh pencahayaan yang berbeda. LBP ampuh untuk mendeskripsikan suatu tekstur, mempunyai daya pembeda yang akurat, dan juga memiliki toleransi terhadap perubahan *gray-scale* yang monotonik (Turiyanto, Purwanto, & Dikairono, 2014).



Gambar 2.3 Jarak dan Banyak Pixel Tetangga yang Terpilih (a)R=1; P=8, (b)R=2; P=16, (c)R=2; P=8

Sumber : Turiyanto, et al (2014)

Operator LBP adalah operator yang merubah gambar menjadi gambar berlabel bilangan bulat yang menggambarkan tampilan *small-scale*. Label yang sering digunakan untuk menggambarkan statistic LBP adala histogram (Pietikäinen, Hadid, Zhao, & Ahonen, 2011). Istilah LBP dimaksudkan sebagai representasi intensitas piksel ketetanggaan dari piksel yang sedang diolah (Turiyanto, Purwanto, & Dikairono, 2014). LBP memiliki nilai jarak untuk menentukan ketetanggaan yang akan dipilih, jarak biasa disimbolkan dengan R dan banyaknya tetangga yang dipilih disimbolkan dengan P . Besar jarak dan ketetanggaan yang dipilih digambarkan dalam Gambar 2.3 (Pietikäinen, Hadid, Zhao, & Ahonen, 2011).

Berikut logika untuk LBP, untuk setiap piksel (p), buat *windowing* 8-bit $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$, dimana $b_i = 0$ jika nilai dari i lebih kecil dari titik pusat (piksel yang sedang diolah) dan bernilai 1 sebaliknya. Fungsi LBP didefinisikan sebagai berikut:

$$LBP_{b,R}(x_c, y_c) = \sum_{b=0}^{b-1} s(g_p - g_c) 2^b \quad (2.1)$$

Fungsi $s(x)$ didefinisikan sebagai berikut :

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

Keterangan :

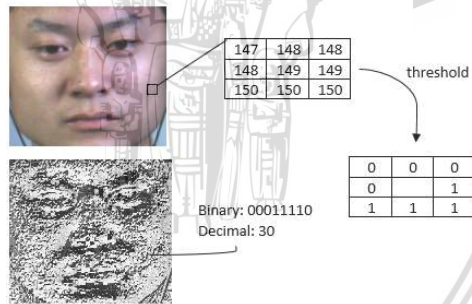
P : jumlah piksel tetangga

R : nilai radius

g_c : nilai piksel x, y

g_p : nilai piksel tetangga

Berikut adalah proses visualisasi perhitungan dari kalkulasi piksel LBP yaitu pada Gambar 2.4.



Gambar 2.4 Proses Kalkulasi Piksel LBP

2.6 Teknologi Pengembangan Sistem

Terdapat beberapa teknologi yang digunakan dalam mengembangkan sistem pengembangan perangkat lunak Layanan Online Kebun Raya Bogor berbasis web, yaitu Codeigniter, OpenCV, Sortsite dan Google Maps API.

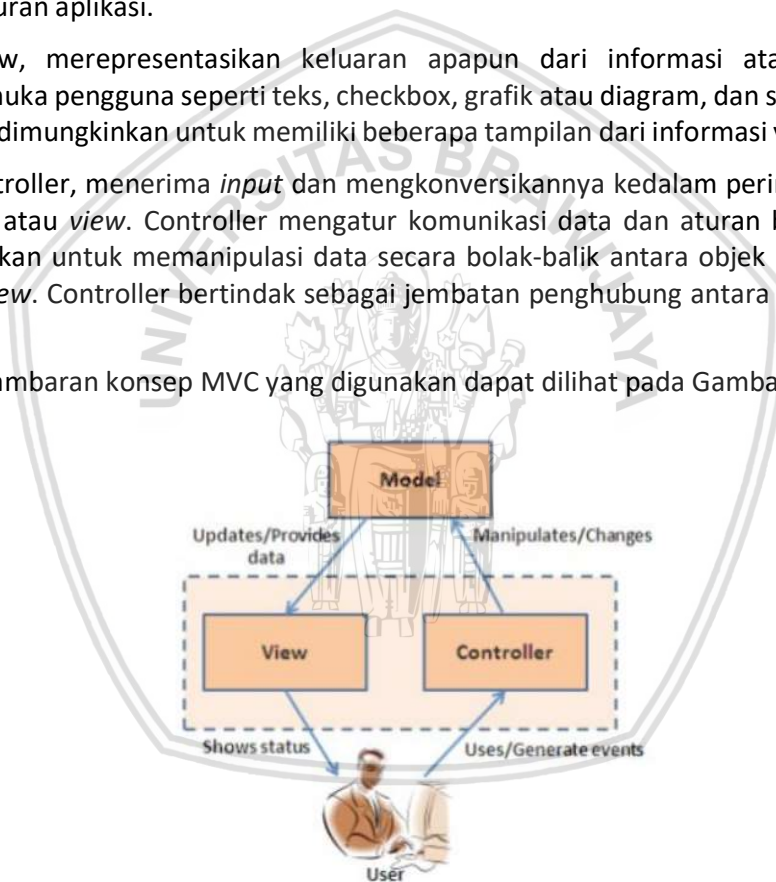
2.6.1 Codeigniter

Codeigniter adalah sebuah *framework* aplikasi *web* yang bersifat *open source* yang digunakan untuk membangun suatu aplikasi dengan bahasa pemrograman PHP. Tujuan utama dari Codeigniter adalah untuk dapat membangun atau mengembangkan suatu aplikasi lebih cepat dari pada harus menulis kode dari awal. *Framework* Codeigniter ini menggunakan pola Model-View-Controller

(MVC) sehingga struktur kode yang dihasilkan lebih terstruktur dan memiliki standar yang jelas. Sesuai dengan namanya, Arsitektur MVC memisahkan aplikasi menjadi tiga bagian, yaitu *Model*, *View* dan *Controller*. MVC digunakan untuk memisahkan akses data dan logika bisnis dari penyajian data dan interaksi pengguna. Pada penelitian yang dilakukan Caytiles (2014), MVC adalah pola arsitektur yang digunakan dalam pengembangan perangkat lunak atau aplikasi. Sebuah aplikasi perangkat lunak dibagi menjadi tiga objek yaitu objek *model*, objek *view*, dan objek *controller*.

1. Model, menangkap perilaku aplikasi dalam domain masalah dan independen dari antarmuka pengguna. Hal ini menandakan informasi atau data dari aplikasi digunakan untuk memanipulasi data dalam arti lain mengelola data, logika data, dan aturan aplikasi.
2. View, merepresentasikan keluaran apapun dari informasi atau elemen antarmuka pengguna seperti teks, checkbox, grafik atau diagram, dan sebagainya. Hal ini dimungkinkan untuk memiliki beberapa tampilan dari informasi yang sama.
3. Controller, menerima *input* dan mengkonversikannya kedalam perintah untuk model atau *view*. Controller mengatur komunikasi data dan aturan bisnis yang digunakan untuk memanipulasi data secara bolak-balik antara objek model dan objek *view*. Controller bertindak sebagai jembatan penghubung antara model dan *view*.

Penggambaran konsep MVC yang digunakan dapat dilihat pada Gambar 2.5.



Gambar 2.5 Hubungan Komponen MVC

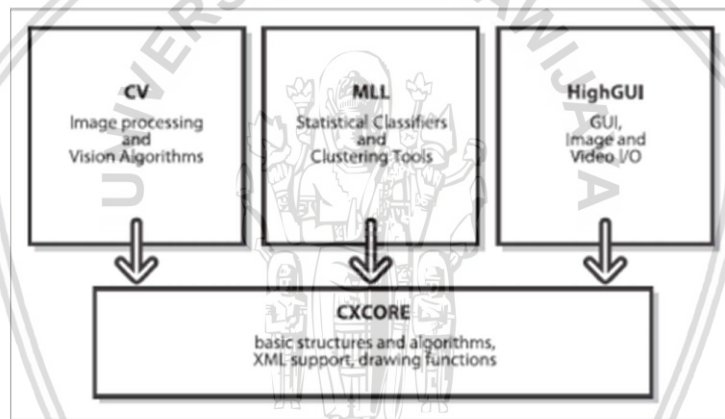
Sumber : Caytiles (2014)

2.6.2 OpenCV

OpenCV adalah sebuah library computer vision yang open source. Library ini ditulis dengan bahas C dan C++ dan dapat dijalankan pada OS Linux, Windows, dan MAC. Saat ini pengembangannya telah dapat berhubungan dengan Python, Ruby, Matlab, dan Bahasa lainnya. OpenCV didesain untuk komputasi yang efisien dan

difokuskan pada aplikasi real time. Salah satu tujuan OpenCV adalah menyediakan infrastruktur *computer vision* yang mudah digunakan yang membantu orang membangun aplikasi citra dengan mudah (Garage, 2008). Library OpenCV mengandung lebih dari 500 function yang menjangkau banyak area termasuk pabrik, inspeksi produk, pencitraan medis, keamanan, user interface, dan robotika. Struktur dari OpenCv terdiri dari lima komponen utama, empat komponen utamanya digambarkan pada Gambar 2.6, yaitu (Gery Bradski, 2008):

1. Komponen CV, terdiri dari pengolahan gambar dasar dan algoritme *computer vision* pada tingkat yang lebih tinggi.
2. MLL, merupakan *library of machine learning* yang meliputi banyak kasifikasi *statistic* dan *clustering tools*.
3. HighGUI mengandung rutin I/O dan *function* mengenai *storing* dan *loading* video dan gambar.
4. CXCore, berisi struktur dan konten data basic.



Gambar 2.6 Struktur OpenCV

(Gery Bradski, 2008)

2.6.3 SortSite

SortSite merupakan salah satu situs yang dapat digunakan sebagai alat pengujian untuk menjalankan ratusan pemeriksaan kualitas pada setiap halaman dari sebuah situs. Setiap halaman dilakukan pengujian kualitas menggunakan lebih dari 450 tempat pemeriksaan (PowerMapper, 2017). Salah satu pengujian yang dilakukan dengan menggunakan SortSite ini adalah pengujian kompatibilitas, pengujian ini bertujuan untuk memastikan bahwa situs dapat berjalan di beberapa peramban berbeda dengan baik. Pengujian kompatibilitas menggunakan SortSite membagi masalah menjadi 3 jenis, yaitu :

1. *Critical Issues* digunakan untuk mengecek fungsionalitas dan kehilangan dari konten web.

2. *Major Issues* digunakan untuk mengecek layout utama seperti letak menu atau navigasi, ukuran gambar dan bentuk tabel.

3. *Minor Issues* digunakan untuk mengecek property pada layout yang ada di halaman seperti fungsi *required* pada *input* dan properti css atau tampilan.

2.6.4 Google Maps API

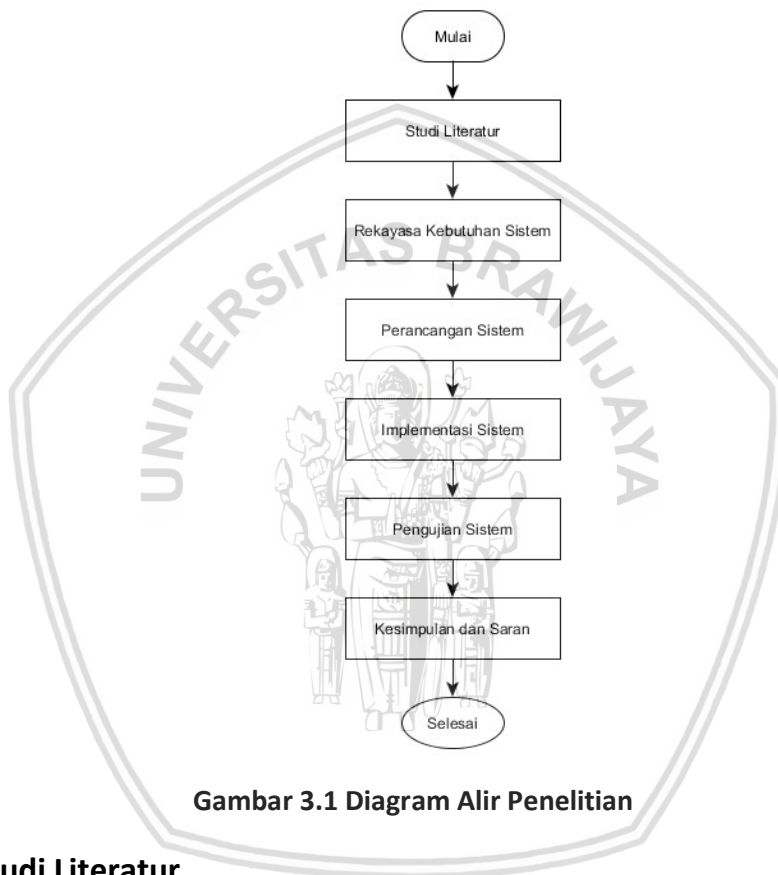
Google Maps merupakan salah satu layanan yang diberikan oleh Google secara gratis yang dapat digunakan untuk melihat peta suatu daerah dengan menggunakan *browser*. Google Maps dapat ditambahkan pada suatu *website* yang dikembangkan dengan memanfaatkan Google Maps API. Google Maps API merupakan suatu *library* yang digunakan untuk menampilkan peta yang berbentuk Javascript (Kindarto, 2008). Google Maps API menyediakan fitur yang berfungsi untuk memanipulasi peta dan dapat menambahkan konten berbagai jenis *services* yang dimiliki. Selain itu penggunaannya dapat diizinkan untuk membangun sebuah aplikasi enterprise di dalam sebuah *website* yang dikembangkan. Untuk cara pemanggilannya dapat dilihat pada Tabel 2.2.

Tabel 2.2 Source Code Menampilkan Peta

Source Code
<pre><script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=true&key=YOUR_API_KEY"></script> <script type="text/javascript"></pre>

BAB 3 METODOLOGI

Metodologi penelitian menjelaskan metode apa saja yang digunakan dalam pengembangan perangkat lunak Layanan Online Kebun Raya Bogor berbasis web. Dalam tahap metodologi penelitian ini menggunakan model *waterfall* dalam proses pembangunan perangkat lunaknya dimana langkah pertama yang dilakukan adalah tahap analisis kebutuhan sistem sampai dengan pengujian sistem. Diagram alir metodologi penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Dasar teori disusun berdasarkan referensi yang diperoleh dari artikel, buku, jurnal, konferensi, serta penelitian-penelitian yang terkait. Studi literatur digunakan untuk mempelajari berbagai referensi sebagai pengetahuan dasar dalam melakukan rekayasa kebutuhan, perancangan, dan implementasi serta pengujian dalam tahap penelitian.

3.2 Rekayasa Kebutuhan Sistem

Rekayasa kebutuhan sistem atau analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem perangkat lunak yang akan dikembangkan. Metode analisis yang digunakan adalah *Object-Oriented Analysis* (OOA) dengan menggunakan bahasa pemodelan *Unified Modeling*

Language (UML). Diagram *use case* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas perangkat lunak dari perspektif *end-user*. Rekayasa kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan perangkat lunak Layanan Online Kebun Raya Bogor yang kemudian akan dimodelkan dalam diagram *use case*. Tiap *use case* dalam diagram *use case* tersebut juga akan dijelaskan lebih rinci dalam *use case scenario*. Pada tahap ini juga dilakukan analisis data yang berfungsi untuk dapat menentukan entitas apa saja yang akan terlibat langsung dengan sistem yang di modelkan dengan *Entity Relationship Diagram* (ERD).

3.3 Perancangan Sistem

Perancangan sistem bertujuan sebagai bahan acuan dalam tahap implementasi dan pengujian perangkat lunak. Pada tahap perancangan sistem ini digunakan sebagai bahan rancangan langkah kerja dari keseluruhan sistem yang akan dikembangkan, baik dari segi model atau segi arsitektur sistem. Tahapan perancangan sistem yang akan dikembangkan adalah sebagai berikut :

1. Perancangan Arsitektur

Pada perancangan arsitektur terdapat diagram *sequence* dan diagram *class*. Diagram *sequence* berguna untuk menggambarkan interaksi antar objek yang terlibat menunjukkan pesan yang dikirim dan kembalian yang diterima, pada bagian ini akan dipaparkan tiga *sequence diagram* yang menggambarkan tiga fungsionalitas utama dari sistem yang dibangun. Diagram *class* berguna untuk menggambarkan *class-class* yang digunakan atau terlibat dalam sistem yang dibangun beserta atribut dan operasinya dan juga menggambarkan relasi antar *class* yang terlibat.

2. Perancangan Komponen

Pada perancangan komponen akan dituliskan tiga sampel algoritme utama yang ada pada sistem yang dibangun. Algoritme yang digunakan akan dituliskan dalam bentuk *pseudocode*.

3. Perancangan Data

Pada perancangan data akan dipaparkan rancangan tabel basis data yang digunakan dalam sistem yang dibangun dalam bentuk PDM (*Physical data Model*) yang akan menjadi dasar dalam implementasi basis data dalam sistem.

4. Perancangan Antarmuka

Pada perancangan antarmuka akan ditampilkan rancangan antarmuka, tata letak komponen yang menjadi dasar dalam implementasi antarmuka sistem. Perancangan antarmuka ini akan menjadi dasar dari implementasi antarmuka sistem.

3.4 Implementasi Sistem

Implementasi sistem adalah fase pembangunan sistem yang mengacu pada perancangan sistem. Proses yang ada pada tahap implementasi ini adalah sebagai berikut :

1. Implementasi Basis Data

Implementasi basis data mengacu pada perancangan basis data pada bagian perancangan data. Pada implementasi data dilakukan pendefinisian tabel yang digunakan dengan *Data Definition Language* (DDL). Implementasi basis data menggunakan Database Management System (DBMS) MySQL.

2. Implementasi Logika Program

Implementasi logika program mengacu pada perancangan komponen pada bagian perancangan sistem yang memuat algoritme-algoritme utama dalam sistem. Logika tersebut diimplementasikan dengan menggunakan bahasa pemrograman PHP dengan *framework* Codeigniter.

3. Implementasi Antarmuka

Implementasi antarmuka mengacu pada perancangan antarmuka yang ada pada bagian perancangan sistem. Implementasi antarmuka ini menggunakan bahasa pemrograman HTML, CSS, dan javascript.

3.5 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibangun telah memenuhi dan berjalan sesuai dengan kebutuhan yang telah didefinisikan. Pengujian juga berfungsi untuk meminimalisir atau menghilangkan masalah atau *bug* yang masih mungkin ada dalam sistem yang dibuat. Ada beberapa tahapan dalam strategi pengujian yang akan dilakukan seperti yang dipaparkan oleh Pressman (2010) yaitu sebagai berikut :

1. Pengujian Unit

Pengujian unit dilakukan untuk menguji unit dalam sistem yang dapat berupa komponen, *class*, atau objek dari sistem berdasarkan perancangan komponen yang ada. Teknik pengujian yang digunakan dalam pengujian unit ini adalah *whitebox testing* dengan metode pengujian *basis path testing*.

2. Pengujian Integrasi

Pengujian integrasi pada sistem menggunakan cara *top-down integration*. Pengujian ini menggunakan teknik *blackbox testing*.

3. Pengujian Validasi

Pengujian validasi dilakukan untuk menguji kesesuaian sistem dengan kebutuhan yang telah ditetapkan. Teknik yang digunakan untuk pengujian validasi ini adalah *blackbox testing*. Pengujian dilakukan dengan melakukan pengujian terhadap sistem sesuai dengan kebutuhan yang telah ditetapkan dengan kondisi masukan yang berikan dan keluaran yang dihasilkan oleh sistem apakah memenuhi kondisi yang diharapkan atau tidak.

4. Pengujian Kompatibilitas

Pengujian kompatibilitas yang akan dilakukan yaitu pada sisi perambah web. Pengujian kompatibilitas dilakukan dengan menggunakan perangkat lunak

bernama SortSite, untuk mengetahui apakah terdapat bagian yang tidak kompatibel dengan perambah web.

3.6 Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah semua tahapan mulai dari studi literatur, analisis kebutuhan, perancangan, implementasi dan pengujian yang diterapkan sudah selesai dilakukan. Selanjutnya dapat diperoleh kesimpulan berdasarkan hasil penelitian yang dilakukan penulis berupa website Layanan Online Kebun Raya Bogor. Kesimpulan dapat diperoleh secara valid berdasarkan hasil yang diperoleh pada tahap pengujian perangkat lunak. Sehingga dapat diperoleh penarikan kesimpulan untuk menjawab perumusan masalah yang telah dirumuskan. Tahap terakhir dalam penulisan yaitu saran. Saran berguna untuk memperbaiki *bug* atau untuk pengembangan sistem selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

Tahapan pertama dalam proses pengembangan sistem adalah rekayasa kebutuhan atau analisis kebutuhan sistem. Pada tahap ini akan menentukan, kebutuhan apa saja yang akan dibutuhkan dalam pengembangan sistem sehingga dapat menentukan sejauh mana sistem yang akan dikembangkan dapat mencapai target yang diharapkan sesuai dengan kebutuhan sistem. Pada tahap analisis kebutuhan nantinya akan menghasilkan pemodelan-pemodelan kebutuhan seperti diagram use case, dan use case scenario yang berguna untuk memudahkan dalam pemahaman mengenai kebutuhan-kebutuhan sistem yang akan dikembangkan.

4.1 Gambaran Umum Sistem

Berdasarkan permasalahan yang melatarbelakangi pengembangan sistem Layanan Online Kebun Raya Bogor ini, maka sistem harus dapat mengetahui kebutuhan-kebutuhan fungsionalitas apa saja yang di butuhkan oleh para stakeholder. Dari hasil observasi yang dilakukan untuk mengetahui domain permasalahan yang ada, didapatkan domain permasalahan tentang bagaimana pihak Pusat Konservasi Tumbuhan Kebun Raya-LIPI melakukan penggalan tingkat kepuasan pengunjung terhadap fasilitas yang sudah disediakan di kebun raya Bogor. Yang kedua yaitu bagaimana pihak PKT Kebun Raya-LIPI dapat meningkatkan pelayanan dengan memberikan kemudahan kepada pengunjung kebun raya Bogor untuk mendapatkan informasi-informasi terkait dengan objek wisata yang ada di kebun raya Bogor. Pengembangan sistem Layanan Online Kebun Raya Bogor berdasarkan dari pengembangan sistem sebelumnya yang memiliki daftar kebutuhan-kebutuhan yang terlampir pada Lampiran C.

4.2 Identifikasi Aktor

Identifikasi aktor ini bertujuan untuk mengidentifikasi aktor-aktor yang nantinya akan berinteraksi dengan sistem secara langsung. Pada Tabel 4.1 terdapat beberapa aktor yang terlibat secara langsung, yaitu *guest*, *member*, dan *admin*.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Guest	Guest adalah pengguna yang belum masuk kedalam sistem.
2	Member	Member adalah pengguna yang telah masuk kedalam sistem.
3	Admin	Admin adalah staff sub bagian kerja sama dan informasi PKT Kebun Raya-LIPI.

4.3 Daftar Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan dari layanan yang harus disediakan, bagaimana sistem bereaksi terhadap sebuah masukan, dan bagaimana sistem harus bertindak untuk sebuah situasi tertentu (Sommerville, 2011). Setiap kebutuhan akan diberikan kode LKRB-1-XX-YY, dimana LKRB menunjukkan nama sistem Layanan Online Kebun Raya Bogor, 1 menunjukkan tipe kebutuhan, 1 untuk kebutuhan fungsional, XX adalah nomor kebutuhan, dan YY adalah spesifikasi kebutuhan. Daftar kebutuhan fungsional dapat dilihat pada Tabel 4.2 dan daftar kebutuhan fungsional dari pengembangan perangkat lunak Layanan Online Kebun Raya yang sebelumnya terdapat pada Lampiran C.

Tabel 4.2 Daftar Kebutuhan Fungsional

a. Guest

No	Kode Kebutuhan	Deskripsi	Nama <i>Use Case</i>
1.	LKRB-1-01	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem yang sesuai dengan otoritasnya.	<i>Login</i>
2.	LKRB-1-02	Sistem harus mampu menampilkan data testimoni pengunjung seperti nama, deskripsi dan foto.	Melihat testimoni
3.	LKRB-1-03	Sistem harus mampu menampilkan peta kebun raya Bogor.	Melihat peta
4.	LKRB-1-04	Sistem harus mampu menampilkan tabel lokasi objek wisata terfavorit.	Melihat daftar objek wisata terfavorit
5.	LKRB-1-05	Sistem harus mampu menampilkan hasil pencarian lokasi objek wisata pada peta kebun raya Bogor	Mencari lokasi objek wisata

b. Member

No	Kode Kebutuhan	Deskripsi	Nama Use Case
1.	LKRB-1-01	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem yang sesuai dengan otoritasnya.	<i>Login</i>
2.	LKRB-1-02	Sistem harus mampu menampilkan data testimoni pengunjung seperti nama, deskripsi dan foto.	Melihat daftar testimoni
3.	LKRB-1-03	Sistem harus mampu menampilkan peta kebun raya Bogor.	Melihat peta
4.	LKRB-1-04	Sistem harus mampu menampilkan tabel lokasi objek wisata terfavorit.	Melihat daftar objek wisata terfavorit
5.	LKRB-1-05	Sistem harus mampu menampilkan hasil pencarian lokasi objek wisata pada peta kebun raya Bogor	Mencari lokasi objek wisata
6.	LKRB-1-06	Sistem harus mampu menyediakan fungsi <i>logout</i> yang berfungsi untuk pengguna yang sudah masuk kedalam sistem sebagai member atau admin dapat keluar dari sistem.	<i>Logout</i>
7.	LKRB-1-07	Sistem harus mampu menyimpan testimoni yang dimasukkan oleh member.	Menyimpan testimoni

c. Admin

No	Kode Kebutuhan	Deskripsi	Nama Use Case
1.	LKRB-1-03	Sistem harus mampu menampilkan peta kebun raya Bogor.	Melihat peta
2.	LKRB-1-05	Sistem harus mampu menampilkan hasil pencarian lokasi objek wisata pada peta kebun raya Bogor	Mencari lokasi objek wisata
3.	LKRB-1-08	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan pengguna	<i>Login admin</i>

		mengakses sistem yang sesuai dengan otoritasnya.	
4.	LKRB-1-09	Sistem harus mampu menyediakan fungsi logout yang berfungsi untuk pengguna yang sudah masuk kedalam sistem sebagai member atau admin dapat keluar dari sistem.	Logout admin
5.	LKRB-1-10	Sistem harus mampu menambah objek wisata baru yang hanya dapat diakses oleh admin.	Menambah objek wisata
6.	LKRB-1-11	Sistem harus mampu mengubah detail informasi objek wisata pada peta kebun raya Bogor.	Mengubah detail informasi objek wisata
7.	LKRB-1-12	Sistem harus mampu menghapus objek wisata yang ada di peta kebun raya Bogor.	Menghapus objek wisata
8.	LKRB-1-13	Sistem harus mampu menampilkan daftar objek wisata dalam bentuk tabel.	Melihat daftar objek wisata
9.	LKRB-1-14	Sistem harus mampu menampilkan grafik tingkat kepuasan pengunjung.	Melihat grafik kepuasan pengunjung

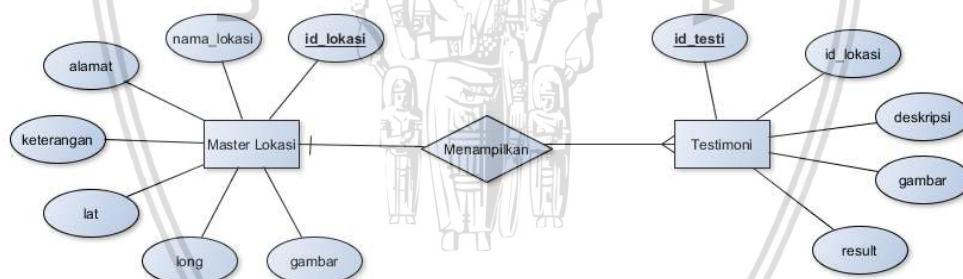
4.3.2 Spesifikasi Kebutuhan

No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
1.	LKRB-1-01	LKRB-1-01-01	Sistem menyediakan <i>form login</i> yang terdiri dari <i>username</i> dan <i>password</i> .
2.	LKRB-1-05	LKRB-1-05-01	Sistem menampilkan <i>marker</i> pada lokasi objek wisata yang dicari.
		LKRB-1-05-02	Sistem menampilkan detail informasi objek wisata seperti nama, alamat, foto dan deskripsi.
3.	LKRB-1-07	LKRB-1-07-01	Sistem menyediakan <i>form</i> untuk menyimpan data testimoni yang terdiri dari nama, lokasi objek wisata, deskripsi dan foto.

4.	LKRB-1-08	LKRB-1-08-01	Sistem menyediakan <i>form login</i> yang terdiri dari <i>username</i> dan <i>password</i> pada halaman admin.
5.	LKRB-1-10	LKRB-1-10-01	Sistem menyediakan <i>form</i> tambah objek wisata yang terdiri dari nama lokasi, keterangan, lintang dan bujur.
6.	LKRB-1-11	LKRB-1-11-01	Sistem menyediakan <i>form</i> ubah detail informasi objek wisata yang terdiri dari nama lokasi, keterangan, lintang dan bujur.

4.4 Analisis Data

Analisis data merupakan langkah yang diambil untuk melakukan analisis terhadap entitas apa saja yang akan terlibat dalam pengembangan sistem. Hasil dari analisis data dapat dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD) pada Gambar 4.1. Pada Gambar 4.1 menggambarkan *Entity Relationship Diagram* (ERD) dari sistem Layanan Online Kebun Raya Bogor. Terlihat entitas, relasi antar entitasnya dan *attribute* pada setiap entitasnya. Untuk ERD pengembangan perangkat lunak Layanan Online Kebun Raya Bogor yang sebelumnya terdapat pada Lampiran F.



Gambar 4.1 ERD Layanan Online Kebun Raya Bogor

4.5 Daftar Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah batasan pada sebuah layanan atau fungsionalitas yang disediakan oleh sistem, kebutuhan non-fungsional biasanya diaplikasikan pada sistem secara keseluruhan, bukan dikenakan pada masing-masing fitur atau layanan (Sommerville, 2011). Setiap kebutuhan akan diberikan kode LKRB-2-XX-YY, dimana LKRB menunjukkan nama sistem Layanan Online Kebun Raya Bogor, 2 menunjukkan tipe kebutuhan, 2 untuk kebutuhan nonfungsional, XX adalah nomor kebutuhan. Daftar kebutuhan non-fungsional dapat dilihat pada Tabel 4.3.

Tabel 4.3 Daftar Kebutuhan Non-Fungsional

No	Kode	Kebutuhan
1	LKRB-2-001	Sistem dapat dijalankan di berbagai jenis browser seperti google chrome, mozilla firefox, dan safari.

4.6 Pemodelan Kebutuhan

4.6.1 Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem dan lingkungannya, atau dapat juga dikatakan *use case diagram* menjelaskan interaksi antara sistem dengan aktor eksternal sistem (Sommerville, 2011). *Use case diagram* dibuat berdasarkan kebutuhan yang telah didefinisikan pada tahap analisis kebutuhan. *Use case diagram* dari sistem yang akan dikembangkan dapat dilihat pada Gambar 4.2. Untuk *use case diagram* perangkat lunak Layanan Online Kebun Raya Bogor pada pengembangan sistem sebelumnya, terdapat pada Lampiran D.



Gambar 4.2 Use Case Diagram Layanan Online Kebun Raya Bogor

4.6.2 Use Case Scenario

Use case scenario digunakan untuk mendeskripsikan mengenai informasi detail *use case*, langkah yang terlibat dalam sebuah *use case* (Miles, 2006).

4.6.2.1 Use Case Scenario Login

Tabel 4.4 merupakan tabel *use case scenario* dari *use case login*.

Tabel 4.4 Use Case Scenario Login

Flow of Events for Login	
Kode Kebutuhan	LKRB-1-001
Objektive	<i>Use case</i> ini berfungsi untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya.
Actor	<i>Guest</i> dan <i>Member</i>
Pre-condition	Aktor telah mengakses halaman <i>login</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form login</i> dengan mengisi <i>username</i> dan <i>password</i>, kemudian memilih tombol <i>login</i>. 2. Sistem menampilkan halaman utama sistem.
Alternative Flow	Apabila <i>username</i> dan <i>password</i> yang dimasukkan tidak terdaftar dalam basis data, maka sistem akan menampilkan pesan " <i>Error! Username atau password yang anda masukan salah</i> ".
Post-condition	Aktor telah masuk kedalam sistem dan dapat mengakses sistem sesuai dengan otoritasnya.

4.6.2.2 Use Case Scenario Melihat Testimoni

Tabel 4.5 merupakan tabel *use case scenario* dari *use case* melihat testimoni.

Tabel 4.5 Use Case Scenario Melihat Testimoni

Flow of Events for Melihat Testimoni	
Kode Kebutuhan	LKRB-1-002
Objektive	<i>Use case</i> ini berfungsi untuk melihat daftar testimoni pengunjung kebun raya Bogor.
Actor	<i>Guest</i> dan <i>Member</i>
Pre-condition	Aktor telah mengakses halaman Layanan Online Kebun Raya Bogor.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu Testimoni. 2. Sistem menampilkan halaman testimoni.

Alternative Flow	-
Post-condition	Aktor melihat daftar testimoni pengunjung kebun raya Bogor.

4.6.2.3 Use Case Scenario Melihat Peta

Tabel 4.6 merupakan tabel *use case scenario* dari *use case* melihat peta.

Tabel 4.6 Use Case Scenario Melihat Peta

Flow of Events for Melihat Peta	
Kode Kebutuhan	LKRB-1-003
Objektive	<i>Use case</i> ini berfungsi untuk melihat peta kebun raya Bogor.
Actor	<i>Guest</i> dan <i>Member</i>
Pre-condition	Aktor telah mengakses halaman Jelajah
Main Flow	Sistem menampilkan halaman jelajah yang terdapat peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi.
Alternative Flow	-
Post-condition	Aktor dapat melihat peta kebun raya Bogor.

4.6.2.4 Use Case Scenario Melihat Daftar Objek Wisata Terfavorit

Tabel 4.7 merupakan tabel *use case scenario* dari *use case* melihat daftar objek wisata terfavorit.

Tabel 4.7 Use Case Scenario Melihat Daftar Objek Wisata Terfavorit

Flow of Events for Melihat Daftar Objek Wisata Terfavorit	
Kode Kebutuhan	LKRB-1-004
Objektive	<i>Use case</i> ini berfungsi untuk melihat daftar objek wisata berdasarkan <i>rating</i> tertinggi.
Actor	<i>Guest</i> dan <i>Member</i>
Pre-condition	Aktor telah mengakses halaman Jelajah.
Main Flow	1. Sistem menampilkan halaman jelajah yang terdapat peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi.
Alternative Flow	-

Post-condition	Aktor dapat melihat daftar objek wisata berdasarkan <i>rating</i> tertinggi.
----------------	--

4.6.2.5 Use Case Scenario Mencari Lokasi Objek Wisata

Tabel 4.8 merupakan tabel *use case scenario* dari *use case* mencari lokasi objek wisata.

Tabel 4.8 Use Case Scenario Mencari Lokasi Objek Wisata

Flow of Events for Mencari Lokasi Objek Wisata	
Kode Kebutuhan	LKRB-1-005
Objektive	<i>Use case</i> ini berfungsi untuk melakukan pencarian lokasi objek wisata pada peta kebun raya Bogor.
Actor	<i>Guest</i> dan <i>Member</i>
Pre-condition	Aktor telah mengakses halaman jelajah.
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman jelajah yang terdapat peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi. 2. Aktor memilih lokasi objek wisata yang dicari. 3. Sistem menampilkan <i>marker</i> pada lokasi objek wisata yang dicari.
Alternative Flow	-
Post-condition	Aktor dapat mengetahui lokasi objek wisata yang dicari.

4.6.2.6 Use Case Scenario Logout

Tabel 4.9 merupakan tabel *use case scenario* dari *use case* *logout*.

Tabel 4.9 Use Case Scenario Logout

Flow of Events for Logout	
Kode Kebutuhan	LKRB-1-006
Objektive	<i>Use case</i> ini berfungsi untuk keluar dari sistem.
Actor	Member
Pre-condition	Aktor telah <i>login</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih <i>logout</i> pada menu profil. 2. Sistem menampilkan halaman utama.
Alternative Flow	-
Post-condition	Aktor berhasil keluar dari otoritasnya.

4.6.2.7 Use Case Scenario Menyimpan Testimoni

Tabel 4.10 merupakan tabel *use case scenario* dari *use case* menyimpan testimoni.

Tabel 4.10 Use Case Scenario Menyimpan Testimoni

Flow of Events for Menyimpan Testimoni	
Kode Kebutuhan	LKRB-1-007
Objektive	<i>Use case</i> ini berfungsi untuk menyimpan testimoni.
Actor	Member
Pre-condition	Aktor telah mengakses halaman testimoni.
Main Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman testimoni yang berisi <i>form</i> untuk menambahkan testimoni dan daftar testimoni pengunjung dan daftar testimoni. 2. Aktor mengisi <i>form</i> testimoni yang terdiri dari nama, lokasi objek wisata, deskripsi dan foto. 3. Sistem menyimpan data pada basis data dan kemudian menampilkan data testimoni terbaru pada halaman testimoni.
Alternative Flow	Apabila aktor mengunggah gambar tidak sesuai dengan tipe file, maka sistem akan menampilkan pesan konfirmasi file gambar yang di unggah tidak sesuai.
Post-condition	Aktor dapat menyimpan testimoni.

4.6.2.8 Use Case Scenario Login Admin

Tabel 4.10 merupakan tabel *use case scenario* dari *use case* login admin.

Tabel 4.11 Use Case Scenario Login Admin

Flow of Events for Login Admin	
Kode Kebutuhan	LKRB-1-008
Objektive	<i>Use case</i> ini berfungsi untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya.
Actor	Admin
Pre-condition	Aktor telah mengakses halaman <i>login</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form login</i> dengan mengisi <i>username</i> dan <i>password</i>, kemudian memilih tombol <i>login</i>. 2. Sistem menampilkan halaman utama sistem.

Alternative Flow	Apabila <i>username</i> dan <i>password</i> yang dimasukkan tidak terdaftar dalam basis data, maka sistem akan menampilkan pesan “Error! Username atau password yang anda masukan salah”.
Post-condition	Aktor telah masuk kedalam sistem dan dapat mengakses sistem sesuai dengan otoritasnya.

4.6.2.9 Use Case Scenario Logout Admin

Tabel 4.12 merupakan tabel *use case scenario* dari *use case logout* admin.

Tabel 4.12 Use Case Scenario Logout Admin

Flow of Events for Logout Admin	
Kode Kebutuhan	LKRB-1-009
Objektive	<i>Use case</i> ini berfungsi untuk keluar dari sistem.
Actor	Admin
Pre-condition	Aktor telah <i>login</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih <i>logout</i> pada menu profil. 2. Sistem menampilkan halaman <i>login</i> admin.
Alternative Flow	-
Post-condition	Aktor berhasil keluar dari otoritasnya.

4.6.2.10 Use Case Scenario Menambah Objek Wisata

Tabel 4.13 merupakan tabel *use case scenario* dari *use case* menambah objek wisata.

Tabel 4.13 Use Case Scenario Menambah Objek Wisata

Flow of Events for Menambah Objek Wisata	
Kode Kebutuhan	LKRB-1-010
Objektive	<i>Use case</i> ini berfungsi untuk menambahkan objek wisata.
Actor	Admin
Pre-condition	Aktor telah mengakses halaman tambah objek wisata.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> pada halaman tambah objek wisata yang terdiri dari nama lokasi, keterangan, lintang dan bujur dan kemudian memilih simpan.

	2. Sistem menyimpan data pada basis data dan menampilkan pesan konfirmasi bahwa data berhasil tersimpan.
Alternative Flow	<p>1. Apabila aktor mengunggah gambar tidak sesuai dengan ketentuan (selain jpg,png dan jpeg) maka sistem akan menampilkan pesan peringatan “File Gambar Tidak Sesuai”.</p> <p>2. Apabila aktor memberikan nama lokasi objek wisata yang akan ditambahkan sama dengan nama lokasi objek wisata yang sudah tersimpan pada database, maka sistem tidak akan menyimpan data dan akan menampilkan pesan peringatan “Nama Lokasi Telah Tersedia”.</p>
Post-condition	Aktor berhasil menambahkan objek wisata baru.

4.6.2.11 Use Case Scenario Mengubah Detail Informasi Objek Wisata

Tabel 4.14 merupakan tabel *use case scenario* dari *use case* mengubah detail informasi objek wisata objek wisata.

Tabel 4.14 Use Case Scenario Mengubah Detail Informasi Objek Wisata

Flow of Events for Mengubah Detail Informasi Objek Wisata	
Kode Kebutuhan	LKRB-1-011
Objektive	<i>Use case</i> ini berfungsi untuk mengubah detail informasi terkait dengan objek wisata.
Actor	Admin
Pre-condition	Aktor telah berada pada halaman edit detail informasi objek wisata.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi <i>form</i> yang terdiri dari nama lokasi, keterangan, gambar, lintang dan bujur. 2. Sistem menyimpan pada database dan akan menampilkan pesan konfirmasi bahwa data berhasil di update.
Alternative Flow	-
Post-condition	Aktor dapat melakukan perubahan data terkait dengan informasi objek wisata.

4.6.2.12 Use Case Scenario Menghapus Objek Wisata

Tabel 4.15 merupakan tabel *use case scenario* dari *use case* menghapus objek wisata.

Tabel 4.15 Use Case Scenario Menghapus Objek Wisata

Flow of Events for Menghapus Objek Wisata	
Kode Kebutuhan	LKRB-1-012
Objektive	<i>Use case</i> ini berfungsi untuk menghapus objek wisata.
Actor	Admin
Pre-condition	Aktor telah mengakses halaman daftar objek wisata.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih objek wisata yang akan dihapus. 2. Sistem menghapus objek wisata yang ada di basis data dan menampilkan kembali daftar objek wisata dengan data terbaru.
Alternative Flow	-
Post-condition	Aktor dapat menghapus objek wisata.

4.6.2.13 Use Case Scenario Melihat Daftar Objek Wisata

Tabel 4.16 merupakan tabel *use case scenario* dari *use case* melihat daftar objek wisata.

Tabel 4.16 Use Case Scenario Melihat Daftar Objek Wisata

Flow of Events for Melihat Daftar Objek Wisata	
Kode Kebutuhan	LKRB-1-013
Objektive	<i>Use case</i> ini berfungsi untuk melihat tabel daftar objek wisata.
Actor	Admin
Pre-condition	Aktor telah <i>login</i> .
Main Flow	<ol style="list-style-type: none"> 1. Aktor telah mengakses halaman daftar objek wisata. 2. Sistem menampilkan tabel daftar objek wisata pada halaman daftar objek wisata.
Alternative Flow	-
Post-condition	Aktor dapat melihat tabel daftar objek wisata.

4.6.2.14 Use Case Scenario Melihat Grafik Kepuasan Pengunjung

Tabel 4.17 merupakan tabel *use case scenario* dari *use case* melihat grafik kepuasan pengunjung.

Tabel 4.17 Use Case Scenario Melihat Grafik Kepuasan Pengunjung

Flow of Events for Melihat Grafik Kepuasan Pengunjung	
Kode Kebutuhan	LKRB-1-014
Objektive	<i>Use case</i> ini berfungsi untuk melihat grafik kepuasan pengunjung.
Actor	Admin
Pre-condition	Aktor telah <i>login</i> .
Main Flow	1. Aktor telah mengakses halaman kelola objek wisata. 2. Sistem menampilkan grafik kepuasan pelanggan.
Alternative Flow	-
Post-condition	Aktor dapat melihat grafik kepuasan pelanggan.



BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

5.1 Perancangan Sistem

Pada proses pengembangan perangkat lunak setelah dilakukan tahap analisis kebutuhan lanjut pada proses berikutnya yaitu perancangan. Dimana setelah kebutuhan di dapatkan pada tahap analisis kebutuhan di modelkan, kemudian masuk pada tahap berikutnya yaitu perancangan. Pada proses pengembangan perangkat lunak Layanan Online Kebun Raya Bogor tahap perancangan dibagi menjadi beberapa bagian, yaitu perancangan arsitektur, perancangan komponen, perancangan data, dan perancangan antarmuka.

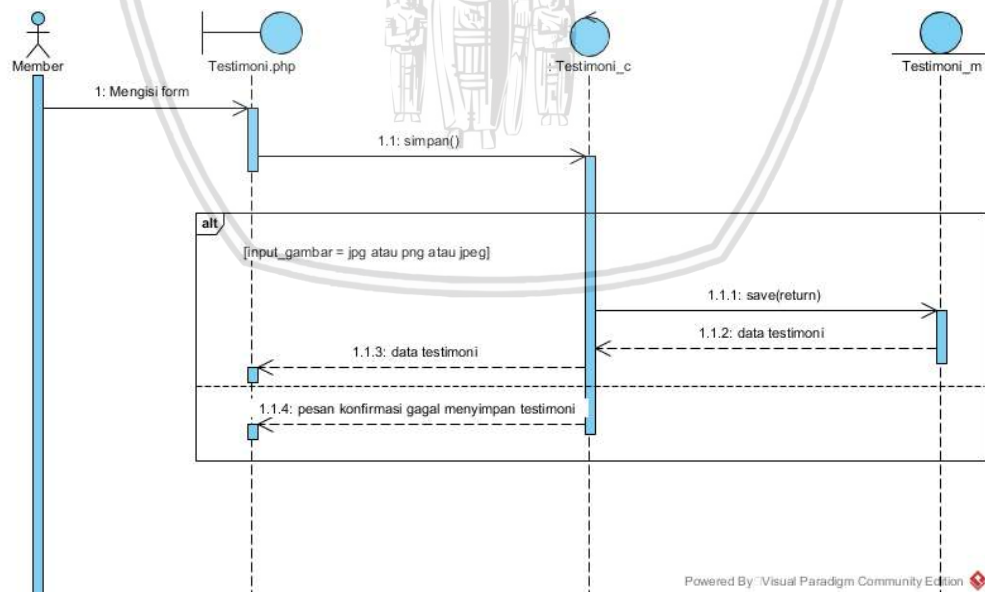
5.1.1 Perancangan Arsitektur

Perancangan arsitektur menjelaskan secara rinci tentang *sequence diagram* dan *class diagram* sistem. *Sequence diagram* berfungsi untuk menjelaskan alur pesan antar objek dalam sistem. Dan *class diagram* digunakan untuk menggambarkan hubungan antar *class* yang saling berinteraksi dalam suatu sistem.

5.1.1.1 Sequence Diagram

Pada bagian ini akan menjelaskan terkait dengan tiga *sequence diagram* yaitu menyimpan testimoni, menambah objek wisata dan mengubah detail informasi objek wisata.

5.1.1.1.1 Sequence Diagram Menyimpan Testimoni

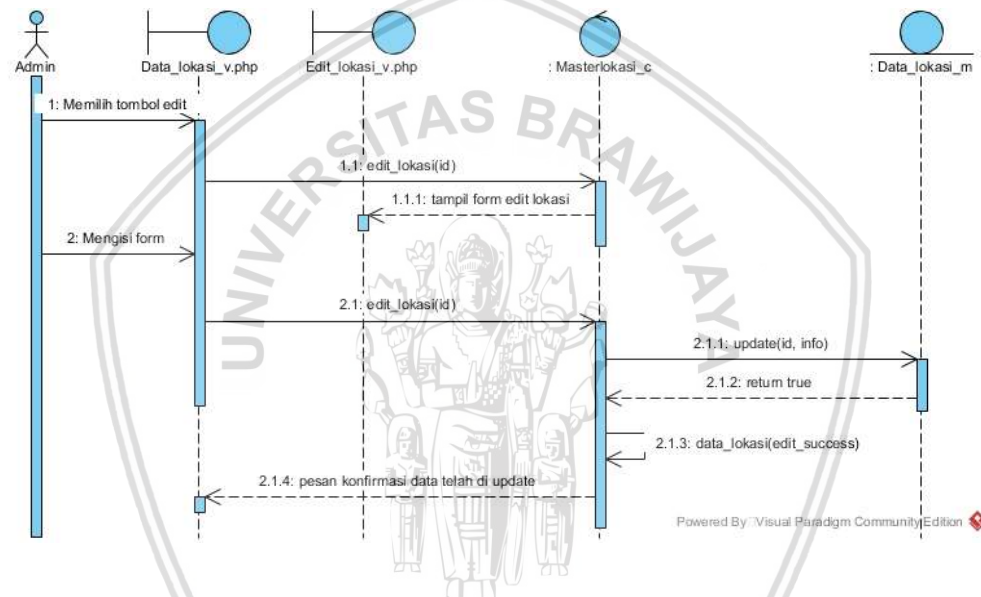


Gambar 5.1 Sequence Diagram Menyimpan Testimoni

Pada Gambar 5.1 merupakan *sequence diagram* menyimpan testimoni yang memiliki *boundary* yaitu Testimoni_v, memiliki objek *controller* yaitu Testimoni_c

dan memiliki satu objek *entity* yaitu Testimoni_m. Untuk menjalankan *use case* menyimpan testimoni, aktor mengisi *form* testimoni pada *boundary* Testimoni_v.php, kemudian *boundary* memanggil *method* `simpan()` pada *controller* Testimoni_c. Kemudian di cek, apabila `input_gambar` yang diunggah oleh aktor memiliki *file extensions* file jpg, png atau jpeg, maka *controller* Testimoni_c akan memanggil *method* `save(return)` untuk menyimpan data testimoni. Jika data testimoni berhasil disimpan maka sistem akan menampilkan pesan konfirmasi data testimoni terbaru pada *boundary* Testimoni_v.php. Jika `input_gambar` memiliki *file extensions* selain file jpg, png dan jpeg, maka sistem akan langsung menampilkan pesan peringatan bahwa file gambar yang diunggah tidak sesuai.

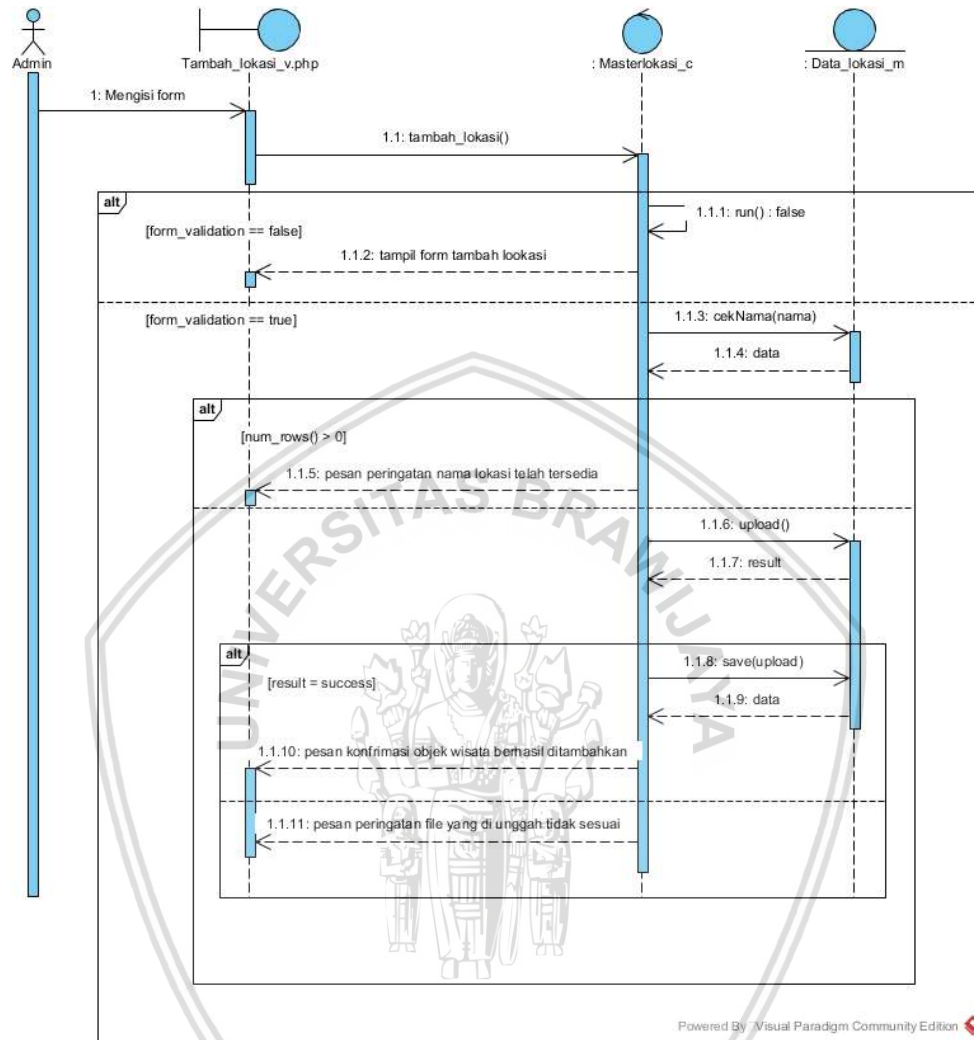
5.1.1.1.2 Sequence Diagram Mengubah Detail Informasi Objek Wisata



Gambar 5.2 Sequence Diagram Mengubah Detail Informasi Objek Wisata

Pada Gambar 5.2 merupakan *sequence diagram* mengubah detail informasi objek wisata yang memiliki *boundary* yaitu Data_lokasi_v dan Edit_lokasi_v, memiliki objek *controller* yaitu Masterlokasi_c dan memiliki objek *entity* yaitu Data_lokasi_m. Untuk menjalankan *use case* mengubah detail informasi objek wisata, aktor yang berperan sebagai *admin* memilih fungsi edit pada halaman edit_lokasi_v. Kemudian *boundary* memanggil *method* `edit_lokasi(id)` pada *controller* Masterlokasi_c untuk menampilkan *form* edit lokasi. Kemudian aktor mengisi *form* pada *boundary* Edit_lokasi_v.php, kemudian memanggil *method* `edit_lokasi(id)` pada *controller* Masterlokasi_c. Untuk melakukan *update* data, maka *controller* memanggil *method* `update(id, info)`, kemudian memanggil *method* `data_lokasi(edit_success)` pada *controller* Masterlokasi_c dan menampilkan pesan konfirmasi data telah berhasil di *update* pada *boundary* Data_lokasi_v.php.

5.1.1.1.3 Sequence Diagram Menambah Objek Wisata



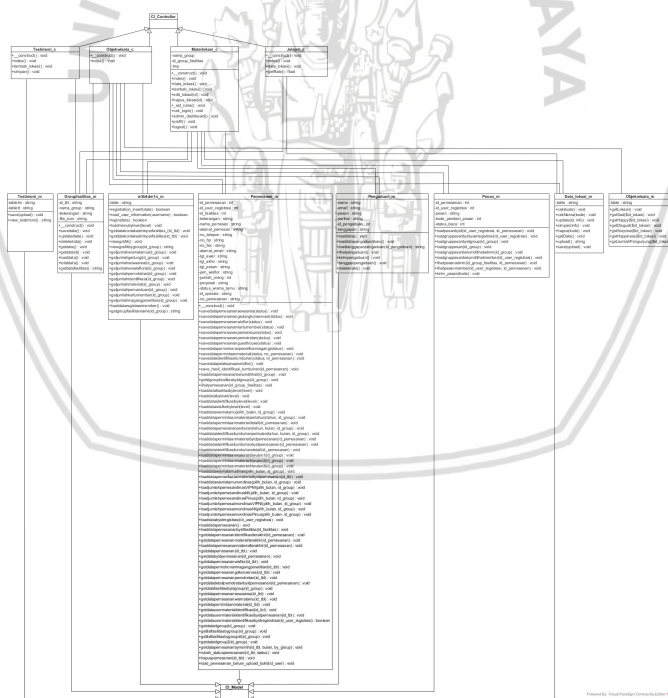
Gambar 5.3 Sequence Diagram Mengubah Detail Informasi Objek Wisata

Pada Gambar 5.3 merupakan *sequence diagram* menambah objek wisata. Untuk menjalankan *use case* menambah objek wisata, admin yang berperan sebagai aktor mengakses *boundary* `tambah_lokasi_v.php` untuk mengisi *form* tambah lokasi objek wisata. Setelah aktor mengisi *form*, *boundary* akan memanggil method `tambah_lokasi()` pada objek *controller* `Masterlokasi_c`. Kemudian di cek `form_validation`, jika `form_validation` bernilai `false`, maka sistem akan menampilkan *form* tambah lokasi. Jika `form_validation` bernilai `true`, maka akan memanggil method `cekNama(nama)` pada objek *entity* `Data_lokasi_m`. Kemudian dilakukan pengecekan jika nama yang dimasukkan oleh aktor sudah tersedia pada *database*, maka sistem akan menampilkan pesan peringatan nama lokasi telah tersedia. Jika nama yang dimasukkan oleh aktor belum tersedia, akan memanggil method `upload()` pada objek *entity* `Data_lokasi_m` dan mengembalikan nilai `result`. Apabila nilai `result = "success"`

akan menyimpan data objek wisata yang akan ditambahkan dengan memanggil method `save(upload)` pada *entity* `Data_lokasi_m`. Jika data berhasil disimpan, maka akan menampilkan pesan konfirmasi bahwa objek wisata berhasil ditambahkan pada *boundary* `tambah_lokasi_v.php`. Jika kembalian *result* tidak bernilai “*success*”, maka sistem akan menampilkan pesan peringatan bahwa file yang di unggah tidak sesuai.

5.1.1.2 Class Diagram

Gambar 5.4 merupakan gambar *class diagram* sistem Layanan Online Kebun Raya Bogor. *Class diagram* berfungsi untuk mengetahui *class-class* apa saja yang membentuk sistem, beserta hubungan antar *class* yang saling berinteraksi satu sama lain. Pada Gambar 5.4 terdapat dua jenis *class*, yaitu *class* yang berperan sebagai *controller* dan *class* yang berperan sebagai *model*. *Class* `CI_Controller` merupakan *general class* untuk *class-class controller* turunannya, yaitu `Testimoni_c`, `Objekwisata_c`, `Masterlokasi_c` dan `Jelajah_c`. Sedangkan *class* `CI_Model` merupakan *general class* untuk *class-class* turunannya yaitu, `Testimoni_m`, `Groupfasilitas_m`, `w3b4dm1n_m`, `Pemesanan_m`, `Pengaduan_m`, `Pesan_m`, `Data_lokasi_m`. Untuk *class diagram* dari pengembangan perangkat lunak Layanan Online Kebun Raya Bogor sebelumnya yaitu dapat digambarkan pada Lampiran E.



Gambar 5.4 *Class Diagram* Layanan Online Kebun Raya Bogor

5.1.2 Perancangan Komponen

Perancangan komponen menjelaskan rincian *sub-sistem* dari setiap komponen pengembangan perangkat lunak. Pada perancangan komponen ini akan mendefinisikan rincian algoritme proses yang terjadi di dalam komponen sistem.

5.1.2.1 Perancangan Komponen *Method* Simpan

Nama Class : Testimoni_c - *Controller*

Nama Operasi : Simpan

Algoritme :

Tabel 5.1 Perancangan Komponen *Method* Simpan

No	Pseudocode
1	data <- array
1	extension <- memanggil method pathinfo dengan parameter input_gambar, name dan PATHINFO_EXTENSION
1	config['upload_path'] = './images/'
1	config['allowed_types'] = 'jpg png jpeg'
1	config['max_size'] = '2048'
1	config['remove_space'] = TRUE
1	Menjalankan library('upload', \$config)
1	Menjalankan upload dengan memanggil initialize(config)
2	If input_gambar memiliki extension jpg, png atau jpeg{
2	upload <- array('file' => upload->data())
2	Memanggil method save yang ada di testimoni_m untuk menyimpan data yang tesimpan pada variable upload
2	Else
3	Menampilkan pesan peringatan "File Gambar yang Anda Masukkan Tidak Sesuai"
3	
4	Kembali menampilkan halaman Testimoni_c

5.1.2.2 Perancangan Komponen *Method* Tambah Lokasi

Nama Class : Masterlokasi_c - *Controller*

Nama Operasi : tambah_lokasi

Algoritme :

Tabel 5.2 Perancangan Komponen *Method* Tambah Lokasi

No	Pseudocode
1	memuat model data lokasi
1	data <- data array
1	id_level <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_level
1	
1	id_group <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_group
1	
1	listmenu <- kemballian dari proses penyimpanan load menu by level dengan data id_level
1	

```

1 pemesananbaru <- kemballian dari proses penyimpanan load
1 data pemesanan belum dilihat dengan data id_group
1 listpesanbelumdilihat <- kemballian dari proses penyimpanan
1 load grup pesan belum dilihat admin dengan data id_group
1 Mengeksekusi proses _set_rules
1     If form_validation bernilai false
2         Menampilkan halaman tambah_lokasi_v
3     Else
3         nama <- nama_lokasi
3         cek <- Memanggil method cekNama dengan
3 parameter nama pada data_lokasi_m
3         if cek->num_rows lebih dari 0
4             Menampilkan pesan "Nama Lokasi
4 Sudah Tersedia"
5             Menampilkan halaman tambah_lokasi_v
5         else
5             upload <- memanggil method upload
pada data_lokasi_m
5             if hasil dari proses method upload
yaitu variable result = "success"
6                 Memanggil method save dengan
6 parameter upload pada data_lokasi_m
6                 Menampilkan pesan konfirmasi
"Lokasi Berhasil Ditambahkan"
6                 Menampilkan halaman
tambah_lokasi_v
7             else
7                 Menampilkan pesan peringatan
8 "File Gambar Tidak Sesuai"
9             Endif
10         Endif
        Endif
    Endif

```

5.1.2.3 Perancangan Komponen *Method* Edit Lokasi

Nama Class : Masterlokasi_c - *Controller*

Nama Operasi : edit_lokasi

Algoritme :

Tabel 5.3 Perancangan Komponen Method Edit Lokasi

No	Pseudocode
1	memuat model data lokasi
2	grafik <- kembalian hasil penyimpanan dari proses view grafik
3	data <- data array
4	id_level <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_level
5	\$id_group <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_group
6	listmenu <- kembalian dari proses penyimpanan load menu by level dengan data id_level
7	pemesananbaru <- kembalian dari proses penyimpanan load data pemesanan belum dilihat dengan data id_group
8	listpesanbelumdilihat <- kembalian dari proses penyimpanan load grup pesan belum dilihat admin dengan data id_group
9	mengeksekusi proses_set_rules
10	If(form_validation bernilai true){
11	info <- data variable nama_lokasi, alamat, keterangan, lat, lng
12	\$this->proses penyimpanan hasil proses update dengan data id dan info
13	memanggil kembali method data_lokasi dengan parameter edit_success pada class Masterlokasi_c
14	Else
15	tampilkan halaman edit_lokasi_v dengan data yang tersimpan pada variable data
16	
17	Endif

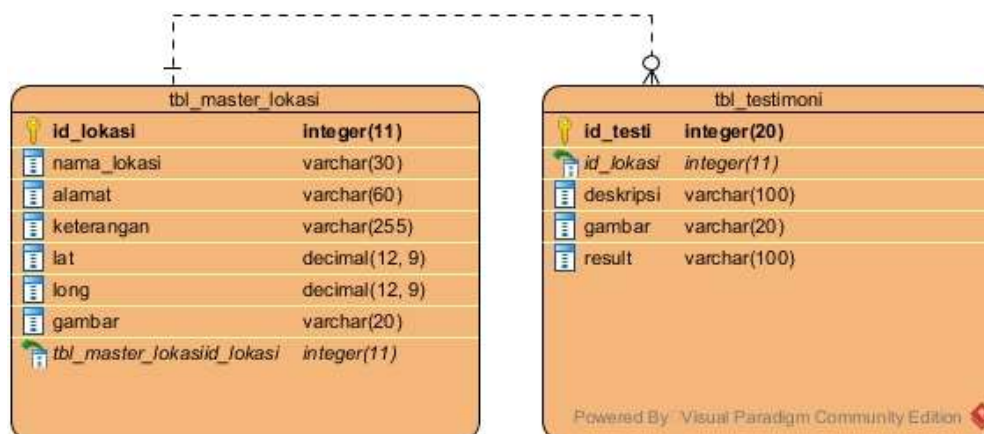
5.1.3 Perancangan Data

Pada Tabel 5.4 merupakan penggambaran terjadinya perubahan pada tahap perancangan data dan implementasi data.

Tabel 5.4 Perubahan Implementasi Data

No	Nama Entitas (analisis)	Nama Entitas (perancangan)
1.	Testimoni	Tbl_testimoni
2.	Master lokasi	Tbl_master_lokasi

Gambar 5.5 merupakan penggambaran *Physical Data Model* (PDM) dari sistem Layanan Online Kebun Raya Bogor.

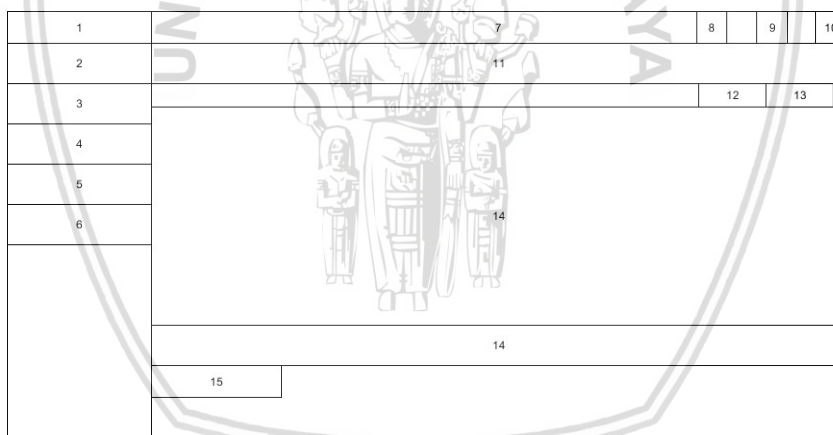


Gambar 5.5 PDM Layanan Online Kebun Raya Bogor

5.1.4 Percangan Antarmuka

Dalam perancangan antarmuka akan digambarkan perancangan antarmuka yang akan diimplementasikan pada sistem.

1. Perancangan Antarmuka Master Lokasi



Gambar 5.6 Perancangan Antarmuka Master Lokasi

Pada Gambar 5.6 merupakan gambaran perancangan antarmuka dari halaman master lokasi dan penjelasan terdapat pada Tabel 5.5.

Tabel 5.5 Perancangan Antarmuka Master Lokasi

No	Nama Objek	Tipe	Keterangan
1	Judul Sistem	Gambar	Judul sistem
2	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
3	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem

4	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
5	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
6	Sidebar Menu	Sidebar Menu	Deretan menu yang terkait dengan sistem
7	Navbar	Navbar	Navbar sistem
8	Pesan	Tombol	Tombol yang digunakan untuk melihat pesan masuk untuk admin
9	Notifikasi	Tombol	Tombol yang digunakan untuk melihat notifikasi masuk untuk admin
10	Informasi Profil	Tombol	Menunjukkan informasi profil
11	Judul Halaman	Text	Menampilkan judul halaman
12	Tambah Lokasi	Tombol	Tombol yang digunakan untuk menambahkan lokasi objek wisata baru
13	Daftar Lokasi	Tombol	Tombol yang digunakan untuk melihat daftar objek wisata.
14	Map	Map	Menampilkan map kebun raya bogor
15	Pilih lokasi objek wisata	Dropdown	Untuk memilih lokasi objek wisata yang akan ditampilkan lokasinya pada peta
16	Submit	Button	Tombol untuk memberikan trigger menampilkan marker pada map berdasarkan lokasi yang sudah dipilih

2. Perancangan Antarmuka Lihat Daftar Objek Wisata

The wireframe illustrates the layout of the 'Lihat Daftar Objek Wisata' page. It includes a top navigation bar (1-10), a left sidebar with a search input (11), a main table area (12) with a table body (13), and a map section (14) with a location dropdown (15) and a submit button (16).

Gambar 5.7 Perancangan Antarmuka Lihat Daftar Objek Wisata

Pada Gambar 5.7 merupakan gambaran perancangan antarmuka dari halaman lihat daftar objek wisata dan penjelasan terdapat pada Tabel 5.6.

Tabel 5.6 Perancangan Antarmuka Lihat Daftar Objek Wisata

No	Nama Objek	Tipe	Keterangan
1	Judul Sistem	Gambar	Judul sistem
2	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
3	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
4	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
5	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
6	Sidebar Menu	Sidebar Menu	Deretan menu yang terkait dengan sistem
7	Navbar	Navbar	Navbar sistem
8	Pesan	Tombol	Tombol yang digunakan untuk melihat pesan masuk untuk admin
9	Notifikasi	Tombol	Tombol yang digunakan untuk melihat notifikasi masuk untuk admin
10	Informasi Profil	Tombol	Menunjukkan informasi profil
11	Judul Halaman	Text	Menampilkan judul halaman
12	Pencarian	-	Kolom untuk melakukan pencarian data.
13	Daftar Lokasi	Tabel	Menampilkan daftar lokasi objek wisata dalam bentuk tabel.
14	Edit	Tombol	Tombol yang digunakan untuk melakukan edit data objek wisata.
15	Delete	Tombol	Tombol yang digunakan untuk menghapus objek wisata.
16	Prev	Tombol	Tombol yang berfungsi untuk melihat tabel pada halaman sebelum halaman yang aktif sekarang.
17	Next	Tombol	Tombol yang berfungsi untuk melihat tabel pada halaman setelah halaman yang aktif sekarang.

3. Perancangan Antarmuka Tambah Objek Wisata

1	7				8	9	10
2	11	12					
3							
4							
5							
6							
	13						
	14						
	15						
	16						
	17						
	18						
					19	20	

Gambar 5.8 Perancangan Antarmuka Tambah Objek Wisata

Pada Gambar 5.8 merupakan gambaran perancangan antarmuka dari halaman tambah objek wisata dan penjelasan terdapat pada Tabel 5.7.

Tabel 5.7 Perancangan Antarmuka Tambah Objek Wisata

No	Nama Objek	Tipe	Keterangan
1	Judul Sistem	Gambar	Judul sistem
2	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
3	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
4	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
5	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
6	Sidebar Menu	Sidebar Menu	Deretan menu yang terkait dengan sistem
7	Navbar	Navbar	Navbar sistem
8	Pesan	Tombol	Tombol yang digunakan untuk melihat pesan masuk untuk admin
9	Notifikasi	Tombol	Tombol yang digunakan untuk melihat notifikasi masuk untuk admin
10	Informasi Profil	Tombol	Menunjukkan informasi profil
11	Halaman Pengelolaan	Tombol	Tombol yang berfungsi untuk menampilkan halaman pengelolaan objek wisata.
12	Map	Map	Menampilkan map kebun raya Bogor.

13	Nama Lokasi	Text	Kolom untuk mengisi nama lokasi objek wisata yang akan ditambahkan.
14	Alamat	Text	Kolom untuk mengisi alamat lokasi objek wisata yang akan ditambahkan.
15	Keterangan	Text	Kolom untuk memberikan keterangan pada lokasi objek wisata yang akan ditambahkan.
16	Lintang	Text	Kolom untuk memberikan titik lintang lokasi objek wisata yang akan ditambahkan.
17	Bujur	Text	Kolom untuk memberikan titik bujur lokasi objek wisata yang akan ditambahkan.
18	Gambar	Tombol	Tombol yang berfungsi untuk menampilkan data dari komputer yang akan dipilih untuk di unggah.
19	Reset	Tombol	Tombol yang berfungsi untuk mengosongkan kembali form yang telah diisi sebelumnya.
20	Submit	Tombol	Tombol yang berfungsi untuk menyimpan form tambah objek wisata.

4. Perancangan Antarmuka Testimoni (*no login*)

Gambar 5.9 Perancangan Antarmuka Testimoni (*no login*)

Pada Gambar 5.9 merupakan gambaran perancangan antarmuka dari halaman Testimoni (*no login*) dan penjelasan terdapat pada Tabel 5.8.

Tabel 5.8 Perancangan Antarmuka Testimoni (no login)

No	Nama Objek	Tipe	Keterangan
1	Logo KRB	Gambar	Gambar logo kebun raya Bogor
2	Logo 2 Abad	Gambar	Gambar logo 2 abad kebun raya Bogor
3	Tanggal	Date	Tanggal hari ini
4	Pencarian	Kolom	Kolom pencarian informasi kebun raya Bogor
5	Menu bar	Menu bar	Berisi barisan menu yang tersedia
6	Judul Halaman	Text	Judul halaman yang sedang aktif yaitu testimoni
7	Tabel testimoni	Tabel	Tabel data testimoni pengunjung
8	Show item	Combo-box	Menampilkan jumlah pilihan data yang ingin ditampilkan
9	Pencarian	Kolom	Pencarian data pada tabel testimoni
10	Penanda halaman tabel	Text	Halaman penanda tabel yang aktif
11	Footer	Text	Footer halaman testimoni

5. Perancangan Antarmuka Testimoni (login)

The diagram illustrates the layout of the Testimoni (login) interface. It features a header section with a logo (1), a date field (3), and a search bar (4). The main content area (5) contains a table (7) with a search bar (9) and a 'Show item' dropdown (8). The table has a page indicator (10). The footer (11) is at the bottom.

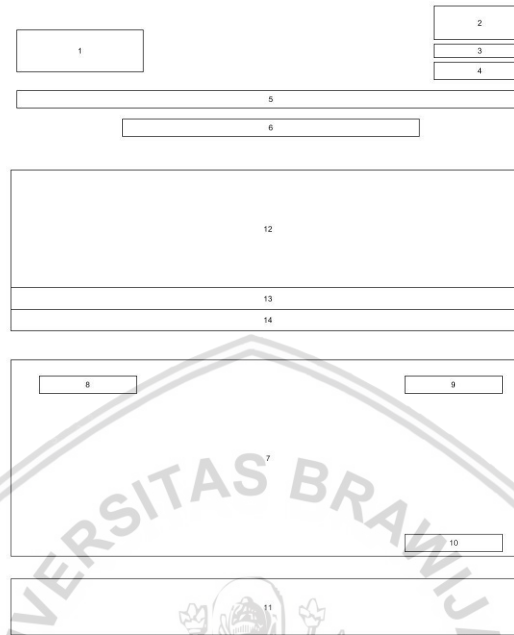
Gambar 5.10 Perancangan Antarmuka Testimoni (login)

Pada gambar 5.10 merupakan gambaran perancangan antarmuka dari halaman Testimoni (*login*) dan penjelasan terdapat pada Tabel 5.9.

Tabel 5.9 Perancangan Antarmuka Testimoni (*login*)

No	Nama Objek	Tipe	Keterangan
1	Logo KRB	Gambar	Gambar logo kebun raya Bogor
2	Logo 2 Abad	Gambar	Gambar logo 2 abad kebun raya Bogor
3	Tanggal	Date	Tanggal hari ini
4	Pencarian	Kolom	Kolom pencarian informasi kebun raya Bogor
5	Menu bar	Menu bar	Berisi barisan menu yang tersedia
6	Judul Halaman	Text	Judul halaman yang sedang aktif yaitu testimoni
7	Tabel testimoni	Tabel	Tabel data testimoni pengunjung
8	Show item	Combo-box	Menampilkan jumlah pilihan data yang ingin ditampilkan
9	Pencarian	Kolom	Pencarian data pada tabel testimoni
10	Penanda halaman tabel	Text	Halaman penanda tabel yang aktif
11	Footer	Text	Footer halaman testimoni
12	Form Testimoni	Form	Form testimoni yang terdiri dari deskripsi, lokasi objek wisata dan gambar

6. Perancangan Antarmuka Jelajah



Gambar 5.11 Perancangan Antarmuka Jelajah

Pada gambar 5.11 merupakan gambaran perancangan antarmuka dari halaman jelajah dan penjelasan terdapat pada Tabel 5.13.

Tabel 5.10 Perancangan Antarmuka Jelajah

No	Nama Objek	Tipe	Keterangan
1	Logo KRB	Gambar	Gambar logo kebun raya Bogor
2	Logo 2 Abad	Gambar	Gambar logo 2 abad kebun raya Bogor
3	Tanggal	Date	Tanggal hari ini
4	Pencarian	Kolom	Kolom pencarian informasi kebun raya Bogor
5	Menu bar	Menu bar	Berisi barisan menu yang tersedia
6	Judul Halaman	Text	Judul halaman yang sedang aktif yaitu testimoni
7	Tabel daftar lokasi objek wisata	Tabel	Tabel data testimoni pengunjung
8	Show item	Combo-box	Menampilkan jumlah pilihan data yang ingin ditampilkan
9	Pencarian	Kolom	Pencarian data pada tabel testimoni

10	Penanda halaman tabel	Text	Halaman penanda tabel yang aktif
11	Footer	Text	Footer halaman testimoni
12	Peta kebun raya Bogor	Map	Menampilkan peta kebun raya Bogor.
13	Lokasi objek wisata	Combo box	Menampilkan lokasi objek wisata kebun raya Bogor.
14	Tampilkan lokasi	Tombol	Memberikan triger untuk melakukan pencarian lokasi objek wisata.

7. Perancangan Antarmuka Halaman Kelola Objek Wisata



Gambar 5.12 Perancangan Antarmuka Kelola Objek Wisata

Pada gambar 5.12 merupakan gambaran perancangan antarmuka dari halaman kelola objek wisata dan penjelasan terdapat pada Tabel 5.13.

Tabel 5.11 Perancangan Antarmuka Kelola Objek Wisata

No	Nama Objek	Tipe	Keterangan
1	Judul Sistem	Gambar	Judul sistem
2	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
3	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
4	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
5	Sidebar Menu	Sidebar	Deretan menu yang terkait dengan sistem
6	Sidebar Menu	Sidebar Menu	Deretan menu yang terkait dengan sistem
7	Navbar	Navbar	Navbar sistem

8	Pesan	Tombol	Tombol yang digunakan untuk melihat pesan masuk untuk admin
9	Notifikasi	Tombol	Tombol yang digunakan untuk melihat notifikasi masuk untuk admin
10	Informasi Profil	Tombol	Menunjukkan informasi profil
11	Grafik Kepuasan Pengunjung	Grafik	Menampilkan grafik kepuasan pengunjung.

5.2 Implementasi Sistem

Proses implementasi dilakukan setelah selesai proses perancangan. Implementasi dilakukan berdasarkan hasil yang diperoleh dari tahap analisis kebutuhan dan tahap perancangan. Setiap kebutuhan yang telah digali dan didapatkan akan diimplementasikan pada sistem. Implementasi harus memuat struktur dan kode program berdasarkan kelas-kelas dan algoritme yang sudah dibuat pada bagian perancangan. Pada bagian implementasi ini akan dijelaskan mengenai spesifikasi sistem, implementasi kode program, implementasi data, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan untuk membangun sistem ini meliputi spesifikasi perangkat keras, perangkat lunak, sistem operasi, dan arsitektur sistem. Sub bab dibawah ini menjelaskan mengenai spesifikasi sistem yang digunakan dalam pengembangan perangkat lunak Layanan Online Kebun Raya Bogor. Tabel 5.12 merupakan tabel spesifikasi dari perangkat lunak yang digunakan dalam pengembangan perangkat lunak Layanan Online Kebun Raya Bogor. Pada Tabel 5.13 merupakan tabel spesifikasi dari perangkat lunak yang digunakan dalam pengembangan perangkat lunak Layanan Online Kebun Raya Bogor. Pada Tabel 5.14 merupakan tabel sistem operasi yang digunakan dalam pengembangan perangkat lunak Layanan Online Kebun Raya Bogor.

Tabel 5.12 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Laptop	<ol style="list-style-type: none"> 1. Intel Core i5 4210U up to 2.7Ghz 2. Memory 4GB 3. Hardisk 500GB

Tabel 5.13 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Editor Dokumentasi	Microsoft Word 2010
Editor Perancangan	Visual Paradigm <i>Community Edition</i> , Data Grip
Editor Pemrograman	Sublime Text 3
Bahasa Pemrograman	PHP, Javascript, CSS, HTML
Framework	Codeigniter
DBMS	MySQL
Browser	Google Chrome

Tabel 5.14 Sistem Operasi

Nama Komponen	Spesifikasi
Sistem Operasi Pengembangan Sistem	Windows 10

5.2.2 Implementasi Kode Program

Implementasi kode program didasarkan pada perancangan komponen. Pada bagian ini akan dirubah algoritme-algoritme yang sudah dibuat dalam bentuk *pseudocode* pada bagian perancangan komponen bentuk bahasa pemrograman.

5.2.2.1 Implementasi Kode Program *Method Simpan*

Nama Class : Testimoni_c - Controller

Nama Operasi : Simpan

Source code :

Tabel 5.15 Implementasi Kode Program *Method Simpan*

No	Source code
1	<code>public function simpan(){</code>
2	<code> \$data = array();</code>
3	<code> \$extension</code>
4	<code> pathinfo(\$_FILES['input_gambar']['name'],</code>
5	<code> PATHINFO_EXTENSION);</code>
6	<code> \$config['upload_path'] = './images/';</code>
7	<code> \$config['allowed_types'] = 'jpg png jpeg';</code>
	<code> \$config['max_size'] = '2048';</code>

8	\$config['remove_space'] = TRUE;
9	\$this->load->library('upload', \$config);
10	\$this->upload->initialize(\$config);
11	if(\$this->upload->do_upload('input_gambar')&&
12	(\$extension == "jpg" \$extension == "png" \$extension == "jpeg")){
13	\$upload = array('file' => \$this->upload->data());
14	\$this->Testimoni_m->save(\$upload);
15	\$this->session->set_flashdata("warning", "<script>alert('Testimoni Berhasil Disimpan');</script>");
16	}else{
17	\$this->session->set_flashdata("warning", "<script>alert('File Gambar yang Anda Masukkan Tidak Sesuai');</script>"); }
18	redirect('Testimoni_c');
19	}

5.2.2.2 Implementasi Kode Program *Method* Tambah Lokasi

Nama Class : Masterlokasi_c - Controller

Nama Operasi : tambah_lokasi

Source code :

Tabel 5.16 Implementasi Kode Program *Method* Tambah Lokasi

No	Source code
1	\$this->load->model('data_lokasi_m');
2	\$data = array();
3	\$id_level = \$this->session->userdata['logged_in']['id_level'];
4	\$id_group = \$this->session->userdata['logged_in']['id_group'];
5	\$data['listmenu'] = \$this->w3b4dm1n_m->loadmenubylevel(\$id_level);
6	\$data['pemesananbaru'] = \$this->pemesanan_m->loaddatapemesananbelumdilihat(\$id_group);
7	\$data['listpesanbelumdilihat'] = \$this->pesan_m->loadgruppesanbelumdilihatadmin(\$id_group);
8	
9	
10	\$this->_set_rules();
11	if(\$this->form_validation->run()==false){
12	\$this->load->view('tambah_lokasi_v',\$data);

```

13         }else{
14             $nama=$this->input->post('nama_lokasi');
15             $cek=$this->data_lokasi_m-
>cekNama($nama);
16             if($cek->num_rows()>0){
17                 $this->session-
>set_flashdata("warning", "<script>alert('Nama Lokasi Sudah
Tersedia')</script>");
18                 $this->load-
>view('tambah_lokasi_v',$data);
19             }else{
20                 $upload = $this->data_lokasi_m-
>upload();
21                 if($upload['result'] == "success"){
22                     $this->data_lokasi_m-
>save($upload);
23                     $this->session-
>set_flashdata("warning", "<script>alert('Objek Lokasi
Berhasil Ditambahkan')</script>");
24                     $this->load-
>view('tambah_lokasi_v',$data);
25                 }else{
26                     $this->session-
>set_flashdata("warning", "<script>alert('File Gambar Tidak
Sesuai')</script>");
27                     $this->load-
>view('tambah_lokasi_v',$data);
28                 }
29             }
30         }

```

5.2.2.3 Implementasi Kode Program *Method* Edit Lokasi

Nama Class : Masterlokasi_c - Controller

Nama Operasi : edit_lokasi

Source code :

Tabel 5.17 Implementasi Kode Program *Method* Edit Lokasi

No	Source code
1	<code>\$this->load->model('data_lokasi_m');</code>
2	<code>\$data['grafik'] = \$this->w3b4dm1n_m->viewgrafik();</code>
3	<code>\$data = array();</code>

```

4  $id_level = $this->session-
   >userdata['logged_in']['id_level'];
5  $id_group = $this->session-
   >userdata['logged_in']['id_group'];
6  $data['listmenu'] = $this->w3b4dm1n_m-
   >loadmenubylevel($id_level);
7  $data['pemesananbaru'] = $this->pemesanan_m-
   >loaddatapemesananbelumdilihat($id_group);
8  $data['registrasibaru'] = $this->w3b4dm1n_m-
   >loaddataregistrasimember();
9  $data['pengaduanbaru'] = $this->pengaduan_m-
   >loaddatapengaduanbaru();
10 $data['listpesanadmin'] = $this->pesan_m-
   >loadgruppesanbyidgroup($id_group);
11 $data['listpesanbelumdilihat'] = $this->pesan_m-
   >loadgruppesanbelumdilihatadmin($id_group);
12 $data['listdata'] = $this->pemesanan_m-
   >loaddatafasilitasbylevel($id_group);
13 $this->_set_rules();
14 if($this->form_validation->run()==true){
15     $info=array(
        'nama_lokasi'=>$this->input-
        >post('nama_lokasi'),
        'alamat'=>$this->input-
        >post('alamat'),
        'keterangan'=>$this->input-
        >post('keterangan'),
        'lat'=>$this->input->post('lat'),
        'long'=>$this->input->post('lng')
    );
16     $this->data_lokasi_m->update($id,$info);
17     redirect('masterlokasi_c/data_lokasi/edit_success');
18     }else{
19         $id = $this->uri->segment(3);
20         $query = $this->data_lokasi_m->cek($id);
21         foreach ($query->result() as $row) {
22             $data['id_lokasi'] = $row->id_lokasi;
23             $data['nama_lokasi'] = $row->nama_lokasi;
24             $data['alamat'] = $row->alamat;
25             $data['keterangan'] = $row->keterangan;
26             $data['lat'] = $row->lat;
27             $data['long'] = $row->long;
28

```


29	}
30	\$this->load->view('edit_lokasi_v',\$data);
31	}

5.2.3 Implementasi Data

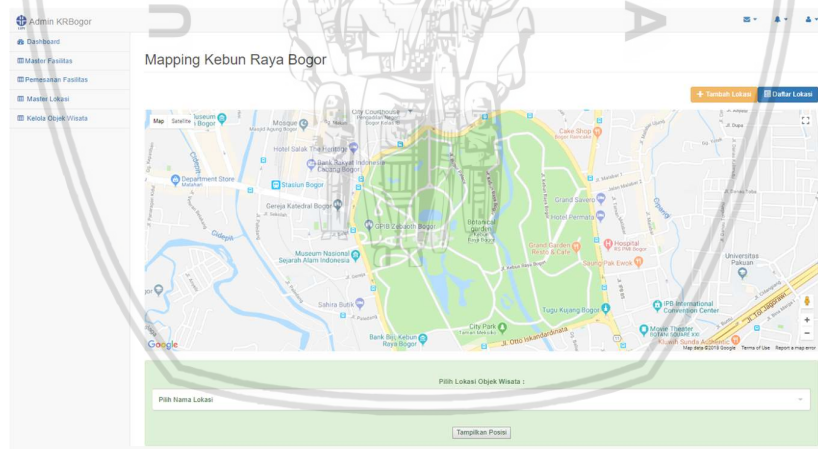
Pada Tabel 5.18 merupakan penggambaran bagaimana terjadinya pembuatan basis data, berdasarkan perancangan PDM.

Tabel 5.18 Implementasi Data Layanan Online Kebun Raya Bogor

No	Data Definition Language
1	Create table tbl_master_lokasi (id_lokasi int(11) primary key, nama_lokasi varchar(200), alamat text, keterangan text, lat decimal(12,9), long decimal(12,9), gambar varchar(100))
2	Create table tbl_testimoni (id_testi int(20), id_lokasi int(11), deskripsi varchar(100), gambar varchar(100), result varchar(100))

5.2.4 Implementasi Antarmuka

1. Implementasi Antarmuka Halaman Master Lokasi



Gambar 5.13 Implementasi Antarmuka Halaman Master Lokasi

Pada Gambar 5.13 merupakan gambaran dari implementasi antarmuka halaman master lokasi berdasarkan perancangan yang dilakukan pada tahapan sebelumnya.

2. Implementasi Antarmuka Halaman Tambah Objek Wisata

Gambar 5.14 Implementasi Antarmuka Halaman Tambah Objek Wisata

Pada Gambar 5.14 merupakan gambaran dari implementasi antarmuka halaman tambah objek wisata berdasarkan perancangan yang dilakukan pada tahapan sebelumnya.

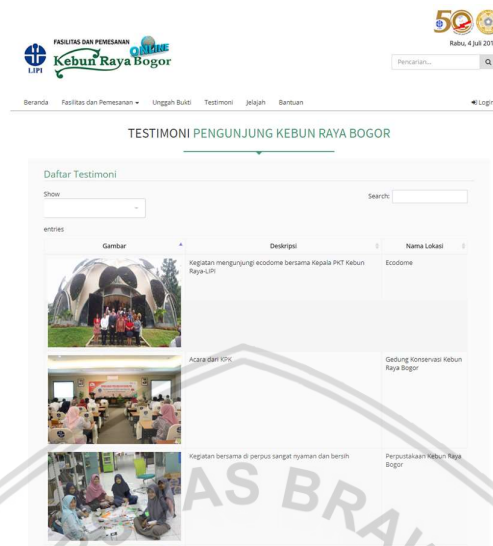
3. Implementasi Antarmuka Halaman Admin Lihat Daftar Objek Wisata

Name Lokasi	Alamat	Keterangan	Lat	Long	Jumlah Testimoni	Aksi
Bunga Bangkai	Jl. Kebun Raya Bogor	Lokasi penempatan bunga bangkai. Tumbuhan bunga bangkai ini seperti dengan apa bunga bangkai yang terlihat di bagian lain. Bunga bangkai memiliki batang yang di bagian bawah. Jadi apabila bunga sedang mekar, maka akan ada bunga bangkai yang di bagian atas.	-6.616122989	101.797324157	6	Detail Hapus
Bunga Raflesia	Jl. Kebun Raya Bogor	Bunga Raflesia	-6.616122152	101.797345614	6	Detail Hapus
Ecodome	Jl. Kebun Raya Bogor	Ecodome merupakan wahana yang di dalamnya terdapat tanaman-tanaman yang di dalam sistemnya juga sebagai pendidikan bagi anak-anak. Selain itu juga, di dalam Ecodome ini terdapat wahana pendidikan mengenai kegiatan pemeliharaan yang berkaitan dengan lingkungan dan objek-objek alam, seni, musik dan budaya.	-6.598140587	101.802246111	7	Detail Hapus
Gedung Konservasi Kebun Raya Bogor	Jl. W. Hudi Djalil No.13, Kota Bogor	Gedung konservasi terdapat di bagian utara kebun raya. Gedung ini memiliki fungsi sebagai tempat penyimpanan koleksi tumbuhan dan hewan yang di koleksi di kebun raya. Gedung ini juga memiliki fungsi sebagai tempat penyimpanan koleksi tumbuhan dan hewan yang di koleksi di kebun raya. Gedung ini juga memiliki fungsi sebagai tempat penyimpanan koleksi tumbuhan dan hewan yang di koleksi di kebun raya.	-6.80261945	101.797326458	2	Detail Hapus

Gambar 5.15 Implementasi Antarmuka Halaman Admin Lihat Daftar Objek Wisata

Pada Gambar 5.15 merupakan gambaran dari implementasi antarmuka halaman lihat daftar objek wisata berdasarkan perancangan yang dilakukan pada tahapan sebelumnya.

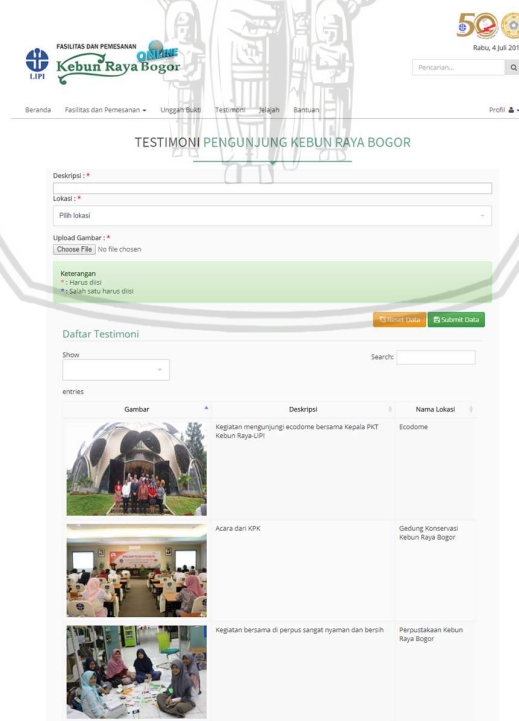
4. Implementasi Antarmuka Halaman Testimoni (*no login*)



Gambar 5.16 Implementasi Antarmuka Halaman Testimoni (*no login*)

Pada Gambar 5.16 merupakan gambaran dari implementasi antarmuka halaman testimoni (*no login*) berdasarkan perancangan yang dilakukan pada tahapan sebelumnya.

5. Implementasi Antarmuka Halaman Testimoni (*login*)



Gambar 5.17 Implementasi Antarmuka Halaman Testimoni (*login*)

Pada Gambar 5.17 merupakan gambaran dari implementasi antarmuka halaman testimoni (*login*) berdasarkan perancangan yang dilakukan pada tahapan sebelumnya.

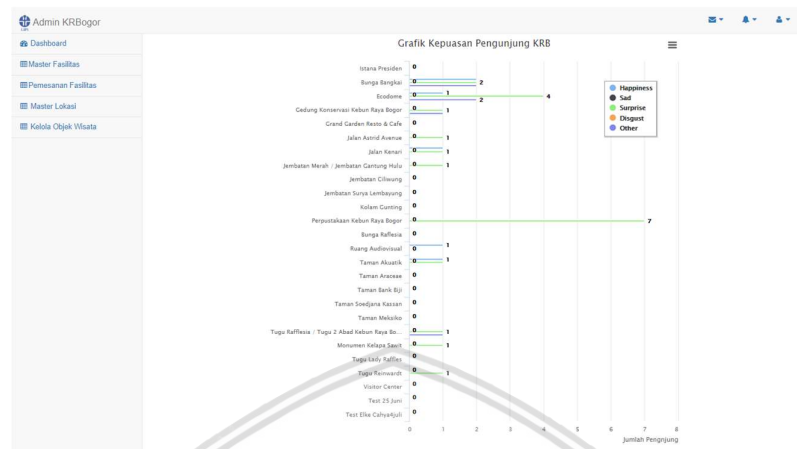
6. Implementasi Antarmukan Halaman Jelajah



Gambar 5.18 Implementasi Antarmuka Halaman Jelajah

Pada Gambar 5.18 merupakan gambaran dari implementasi antarmuka halaman jelajah.

7. Implementasi Antarmuka Halaman Pengelolaan Objek Wisata



Gambar 5.19 Implementasi Antarmuka Halaman Pengelolaan Objek Wisata

Pada Gambar 5.19 merupakan gambaran dari implementasi antarmuka halaman pengelolaan objek wisata.

BAB 6 PENGUJIAN SISTEM

6.1 Pengujian Unit

Pengujian unit berkonsentrasi pada setiap unit (komponen) dari perangkat lunak yang diimplementasi. Pada pengujian unit ini akan menguji 3 method utama yaitu method simpan, tambah lokasi dan edit lokasi.

6.1.1 Pengujian Unit *Method* Simpan

1. *Pseudocode*

Nama Class : Testimoni_c - *Controller*

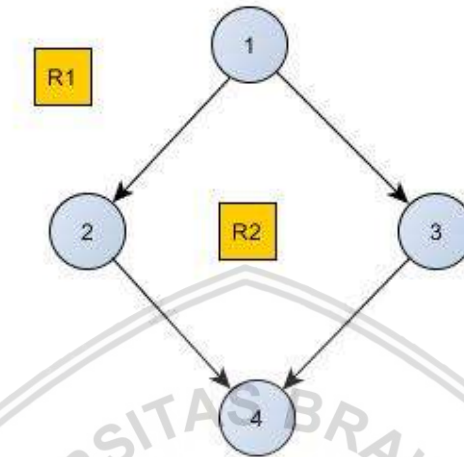
Nama Operasi : Simpan

Tabel 6.1 *Pseudocode* Pengujian Unit Method Simpan

No	<i>Pseudocode</i>
1	data <- array
1	extension <- memanggil method pathinfo dengan parameter input_gambar, name dan PATHINFO_EXTENSION
1	config['upload_path'] = './images/'
1	config['allowed_types'] = 'jpg png jpeg'
1	config['max_size'] = '2048'
1	config['remove_space'] = TRUE
1	Menjalankan library('upload', \$config)
1	Menjalankan upload dengan memanggil initialize(config)
2	If input_gambar memiliki extension jpg, png atau jpeg{
2	upload <- array('file' => upload->data())
2	Memanggil method save yang ada di testimoni_m untuk menyimpan data yang tesimpan pada variable upload
2	Else
3	Menampilkan pesan peringatan "File Gambar yang Anda Masukkan Tidak Sesuai"
3	
4	Kembali menampilkan halaman Testimoni_c

2. Basis Path Testing

2.1 Flow Graph



Gambar 6.1 Flow Graph Method Simpan

2.2 Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 2$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 4 - 4 + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3 Independent Path

- Jalur 1 : 1-2-4
- Jalur 2 : 1-3-4

Tabel 6.2 Hasil Pengujian Unit Mim stillethod Simpan

No	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	1	Jika file extension pada variable input_gambar = jpg, png atau jpeg.	Data berhasil tersimpan dan menampilkan kembali halaman testimoni.	Data berhasil tersimpan dan menampilkan kembali halaman testimoni.	Valid
2	2	Jika file extension pada variable input_gambar selain jpg, png atau jpeg.	Sistem akan menampilkan pesan "File Gambar yang Anda	Sistem akan menampilkan pesan "File Gambar yang Anda	Valid

			Masukkan Tidak Sesuai".	Masukkan Tidak Sesuai".	
--	--	--	----------------------------	----------------------------	--

6.1.2 Pengujian Unit Method Tambah Lokasi

1. Pseudocode

Nama Class : Masterlokasi_c - Controller

Nama Operasi : tambah_lokasi

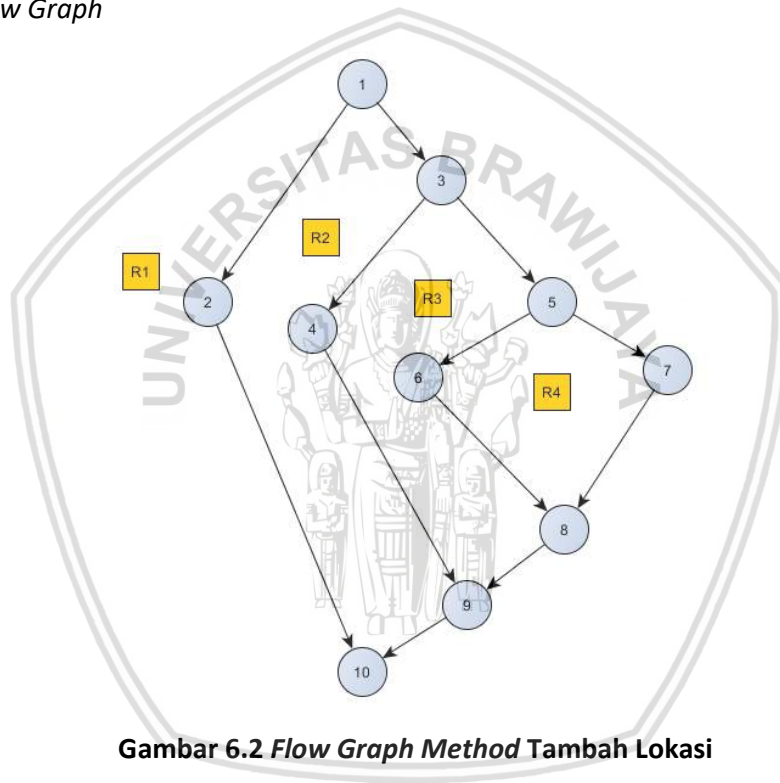
Tabel 6.3 Pseudocode Pengujian Unit Method Tambah Lokasi

No	Pseudocode
1	memuat model data lokasi
1	data <- data array
1	id_level <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_level
1	id_group <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_group
1	listmenu <- kemballian dari proses penyimpanan load menu by level dengan data id_level
1	pemesananbaru <- kemballian dari proses penyimpanan load data pemesanan belum dilihat dengan data id_group
1	listpesanbelumdilihat <- kemballian dari proses penyimpanan load grup pesan belum dilihat admin dengan data id_group
1	Mengeksekusi proses _set_rules
1	If form_validation bernilai false
2	Menampilkan halaman tambah_lokasi_v
3	Else
3	nama <- nama_lokasi
3	cek <- Memanggil method cekNama dengan parameter nama pada data_lokasi_m
3	if cek->num_rows lebih dari 0
4	Menampilkan pesan "Nama Lokasi Sudah Tersedia"
4	Menampilkan halaman tambah_lokasi_v
5	else
5	upload <- memanggil method upload pada data_lokasi_m
5	if hasil dari proses method upload yaitu variable result = "success"
6	Memanggil method save dengan parameter upload pada data_lokasi_m
6	Menampilkan pesan konfirmasi "Lokasi Berhasil Ditambahkan"

6	tambah_lokasi_v	Menampilkan halaman
7	else	
7	"File Gambar Tidak Sesuai"	Menampilkan pesan peringatan
8	Endif	
9	Endif	
10	Endif	

2. Basis Path Testing

2.1 Flow Graph



Gambar 6.2 Flow Graph Method Tambah Lokasi

2.2 Cyclomatic Complexity ($V(G)$)

- $V(G)$ = jumlah *region* = 4
- $V(G)$ = jumlah *edge* - jumlah *node* + 2 = 12 - 10 + 2 = 4
- $V(G)$ = jumlah *predicate node* + 1 = 3 + 1 = 4

2.3 Independent Path

- Jalur 1 : 1-2-10
- Jalur 2 : 1-3-4-9-10
- Jalur 3 : 1-3-5-6-8-9-10
- Jalur 4 : 1-3-5-7-8-9-10

Tabel 6.4 Hasil Pengujian Unit Method Tambah Lokasi

No	No. Jalur	Data Pengujian	Expected Result	Result	Status
1.	1	Jika form_validation bernilai <i>false</i> .	Menampilkan halaman tambah lokasi.	Menampilkan halaman tambah lokasi.	<i>Valid</i>
2.	2	Jika form_validation bernilai <i>true</i> dan nilai dari numrows > 0.	Menampilkan pesan "Nama Lokasi Sudah Tersedia" pada halaman tambah objek wisata.	Menampilkan pesan "Nama Lokasi Sudah Tersedia" pada halaman tambah objek wisata.	<i>Valid</i>
3.	3	Jika form_validation bernilai <i>true</i> , nilai dari numrows tidak lebih dari 0 dan nilai result = " <i>success</i> ".	Data berhasil tersimpan dan menampilkan pesan "Objek Lokasi Berhasil Ditambahkan" pada halaman tambah objek wisata.	Data berhasil tersimpan dan menampilkan pesan "Objek Lokasi Berhasil Ditambahkan" pada halaman tambah objek wisata.	<i>Valid</i>
4.	4	Jika form_validation bernilai <i>true</i> , nilai dari numrows tidak lebih dari 0 dan nilai result = " <i>failed</i> ".	Menampilkan pesan "File Gambar Tidak Sesuai" pada halaman tambah objek wisata.	Menampilkan pesan "File Gambar Tidak Sesuai" pada halaman tambah objek wisata.	<i>Valid</i>

6.1.3 Pengujian Unit *Method* Edit Lokasi

1. *Pseudocode*

Nama Class : Masterlokasi_c - *Controller*

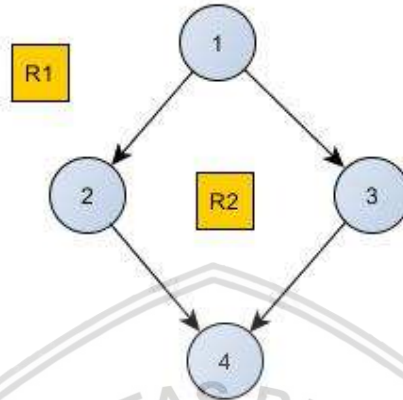
Nama Operasi : edit_lokasi

Tabel 6.5 *Pseudocode* Pengujian Unit *Method* Edit Lokasi

No	<i>Pseudocode</i>
1	memuat model data lokasi
1	grafik <- kembalian hasil penyimpanan dari proses view grafik
1	data <- data array
1	id_level <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_level
1	\$id_group <- hasil pengecekan session dari userdata dengan parameter logged_in dan id_group
1	listmenu <- kembalian dari proses penyimpanan load menu by level dengan data id_level
1	pemesananbaru <- kembalian dari proses penyimpanan load data pemesanan belum dilihat dengan data id_group
1	listpesanbelumdilihat <- kembalian dari proses penyimpanan load grup pesan belum dilihat admin dengan data id_group
1	mengeksekusi proses _set_rules
1	If(form_validation bernilai true){
2	info <- data variable nama_lokasi, alamat, keterangan, lat, lng
2	\$this->proses penyimpanan hasil proses update dengan data id dan info
2	memanggil kembali method data_lokasi dengan parameter edit_success pada class Masterlokasi_c
3	Else
3	tampilkan halaman edit_lokasi_v dengan data yang tersimpan pada variable data
4	Endif

2. Basis Path Testing

2.1 Flow Graph



Gambar 6.3 Flow Graph Method Edit Lokasi

2.2 Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 2$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 4 - 4 + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

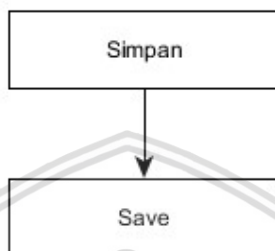
2.3 Independent Path

- Jalur 1 : 1-2-3
- Jalur 2 : 1-3-4

No	No. Jalur	Data Pengujian	Expected Result	Result	Status
1.	1	Jika <code>form_validation</code> bernilai <i>true</i> .	Menampilkan pesan "Data berhasil di simpan".	Menampilkan pesan "Data berhasil di simpan".	<i>Valid</i>
2.	2	Jika <code>form_validation</code> bernilai <i>false</i> .	Menampilkan halaman edit lokasi objek wisata.	Menampilkan halaman edit lokasi objek wisata.	<i>Valid</i>

6.2 Pengujian Integrasi

Pengujian integrasi berfokus pada desain dan konstruksi dari arsitektur perangkat lunak. Pengujian integrasi diterapkan pada proses yang mengintegrasikan fungsionalitas dari beberapa *class* atau *method* untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi ini digunakan pendekatan *top down*, dapat digambarkan pada Gambar 6.4 hubungan antara *method* simpan dan *method* save.



Gambar 6.4 Diagram Hierarki Pengujian Integrasi *Method* Simpan

Tabel 6.6 merupakan tabel langkah uji dari pengujian integrasi yang akan dilakukan.

Tabel 6.6 Langkah Uji Pengujian Integrasi

No	Langkah Uji	Keterangan
1	Simpan() + Save()	<i>Method</i> simpan digunakan untuk menyimpan testimoni yang di masukan oleh aktor. <i>Method</i> simpan yang berada pada <i>class</i> <i>controller</i> dijalankan untuk menguji <i>method</i> <i>stub_save</i> pada <i>class</i> <i>model</i> . Diberikan nilai masukan yang bertipe data <i>string</i> yaitu yang menunjukan kondisi benar adalah "Testimoni Berhasil Disimpan" dan kondisi salah "Testimoni Gagal Disimpan" yang bernilai <i>false</i> .

Pada langkah pertama pengujian integrasi antara *method* *simpan* dan *save* menggunakan *stub*. *Stub* yang akan digunakan yaitu *method* *stub_save* pada *class* *Testimoni_m*. *Method* *stub_save* akan diberikan dua kondisi yaitu kondisi benar bertipe data *string* "Testimoni Berhasil Disimpan" dan kondisi salah bertipe data *string* "Testimoni Gagal Disimpan". Pada pengujian pertama akan dijalankan kondisi benar dengan diberikan masukan yang bernilai `input_deskripsi = "Testimoni Berhasil Disimpan"`. Kode program untuk *method* *simpan* pada percobaan kondisi benar terdapat pada Tabel 6.7.

Tabel 6.7 Source Code Method Simpan Kondisi Benar

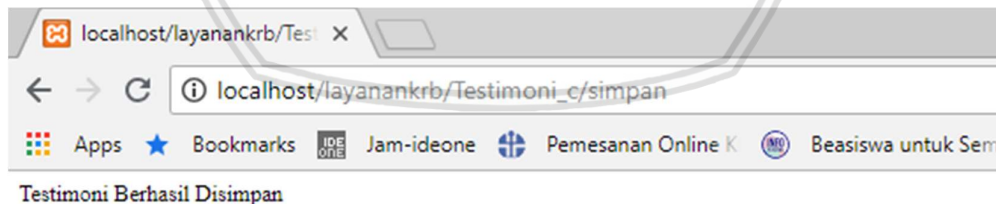
1	public function simpan(){
2	\$input_deskripsi = "Testimoni Berhasil Disimpan";
3	if(\$input_deskripsi == "Testimoni Berhasil Disimpan"){
4	\$result = \$this->Testimoni_m->stub_save(\$input_deskripsi);
5	echo \$result;
6	}else{
7	echo \$input_deskripsi;
8	}
9	}

Kode program untuk *method* `stub_save` yang akan digunakan, terdapat pada Tabel 6.8.

Tabel 6.8 Source Code Method Stub_save

1	public function stub_save(\$upload){
2	return \$upload;
3	}

Hasil dari pengujian pertama dengan di berikan masukkan pada *variable* `input_deskripsi = "Testimoni Berhasil Disimpan"` berhasil mengeksekusi kondisi jika `input_deskripsi` bernilai `Testimoni Berhasil Disimpan`, maka akan memanggil *method* `stub_save` dan akan menampilkan pesan "Testimoni Berhasil Disimpan". Dibuktikan dengan Gambar 6.5 hasil pengujian *method* `stub_save` dengan kondisi benar.

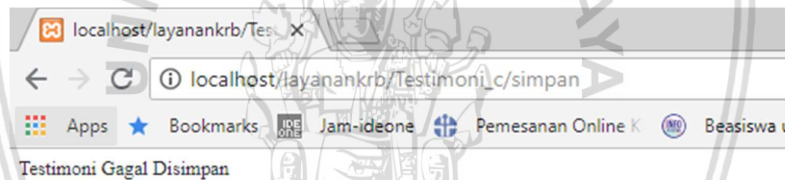
**Gambar 6.5 Hasil Pengujian Method stub_save Kondisi Benar**

Pengujian kedua dilakukan dengan kondisi apabila `input_deskripsi = "Testimoni Gagal Disimpan"`. Jika dilakukan pengecekan `input_deskripsi` bernilai salah, maka akan menampilkan pesan "Testimoni Gagal Disimpan". Kode program untuk *method* `simpan` dengan kondisi salah terdapat pada Tabel 6.9.

Tabel 6.9 Source Code Method Simpan Kondisi Salah

1	public function simpan() {
2	\$input_deskripsi = "Testimoni Berhasil Disimpan";
3	if(\$input_deskripsi == "Testimoni Berhasil Disimpan") {
4	\$result = \$this->Testimoni_m->stub_save(\$input_deskripsi);
5	echo \$result;
6	} else {
7	echo \$input_deskripsi;
8	}
9	}

Hasil dari pengujian kedua di berikan masukkan pada *variable* `input_deskripsi = "Testimoni Gagal Disimpan"` tidak berhasil mengeksekusi kondisi jika `input_deskripsi` bernilai Testimoni Berhasil Disimpan, maka akan menampilkan pesan "Testimoni Gagal Disimpan". Dibuktikan dengan Gambar 6.6 hasil pengujian method `stub_save` dengan kondisi benar.

**Gambar 6.6 Hasil Pengujian Method `stub_save` Kondisi Salah**

6.3 Pengujian Validasi

Pengujian validasi dilakukan untuk memeriksa apakah semua kebutuhan yang ditetapkan sudah dibangun dan apakah sesuai dengan skenario yang telah dibuat. Pengujian validasi dikatakan berhasil apabila sistem yang telah diimplementasi dapat menyediakan semua kebutuhan dan alurnya sesuai dengan skenario yang telah ditetapkan.

Tabel 6.10 merupakan tabel untuk pengujian validasi *use case login*.

Tabel 6.10 Pengujian Validasi *Login*

Kode Kebutuhan	LKRB-1-01
Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman <i>login</i>. 2. Aktor mengisi <i>form login</i> pada halaman login dengan mengisi <i>username</i> = tiara dan <i>password</i> = t14r4, kemudian memilih tombol <i>login</i>.

Expected Result	Sistem menampilkan halaman utama sistem.
Result	Sistem menampilkan halaman utama sistem.
Status	<i>Valid</i>

Tabel 6.11 merupakan tabel untuk pengujian validasi *use case login* alternatif pertama.

Tabel 6.11 Pengujian Validasi Login Alternatif 1

Kode Kebutuhan	LKRB-1-01
Nama Kasus Uji	<i>Login</i>
Prosedur	1. Mengakses halaman <i>login</i> . 2. Aktor mengisi <i>form login</i> dan username tidak terdaftar pada basis data.
Expected Result	Sistem menampilkan pesan “Error! Username atau password yang anda masukan salah”.
Result	Sistem menampilkan pesan “Error! Username atau password yang anda masukan salah”.
Status	Valid

Tabel 6.12 merupakan tabel untuk pengujian validasi *use case* melihat testimoni.

Tabel 6.12 Pengujian Validasi Melihat Testimoni

Kode Kebutuhan	LKRB-1-02
Nama Kasus Uji	Melihat Testimoni
Prosedur	Mengakses halaman testimoni.
Expected Result	Sistem menampilkan daftar testimoni.
Result	Sistem menampilkan daftar testimoni.
Status	Valid

Tabel 6.13 merupakan tabel untuk pengujian validasi *use case* melihat peta.

Tabel 6.13 Pengujian Validasi Melihat Peta

Kode Kebutuhan	LKRB-1-03
Nama Kasus Uji	Melihat Peta
Prosedur	Mengakses halaman jelajah.

Expected Result	Sistem menampilkan peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi objek wisata pada halaman jelajah.
Result	Sistem menampilkan peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi objek wisata pada halaman jelajah.
Status	-

Tabel 6.14 merupakan tabel untuk pengujian validasi *use case* melihat daftar objek wisata terfavorit.

Tabel 6.14 Pengujian Validasi Melihat Daftar Objek Wisata Terfavorit

Kode Kebutuhan	LKRB-1-04
Nama Kasus Uji	Melihat Daftar Objek Wisata Terfavorit
Prosedur	Mengakses halaman jelajah.
Expected Result	Sistem menampilkan peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi objek wisata pada halaman jelajah.
Result	Sistem menampilkan peta kebun raya Bogor, tabel peringkat lokasi objek wisata terfavorit dan <i>form</i> pencarian lokasi objek wisata pada halaman jelajah.
Status	-

Tabel 6.15 merupakan tabel untuk pengujian validasi *use case* mencari lokasi objek wisata.

Tabel 6.15 Pengujian Validasi Mencari Lokasi Objek Wisata

Kode Kebutuhan	LKRB-1-05
Nama Kasus Uji	Mencari Lokasi Objek Wisata
Prosedur	1. Mengakses halaman jelajah. 2. Memilih objek wisata yang dicari pada menu dropdown 3. Select objek wisata kemudian pilih "Tampilkan lokasi"
Expected Result	Sistem menampilkan <i>marker</i> pada lokasi objek wisata yang dicari.
Result	Sistem menampilkan <i>marker</i> pada lokasi objek wisata yang dicari.
Status	Valid

Tabel 6.16 merupakan tabel untuk pengujian validasi *use case logout*.

Tabel 6.16 Pengujian Validasi Logout

Kode Kebutuhan	LKRB-1-06
Nama Kasus Uji	Logout
Prosedur	Memilih <i>logout</i> pada menu profil.
Expected Result	Sistem menampilkan halaman utama.
Result	Sistem menampilkan halaman utama.
Status	Valid

Tabel 6.17 merupakan tabel untuk pengujian validasi *use case* menyimpan testimoni.

Tabel 6.17 Pengujian Validasi Menyimpan Testimoni

Kode Kebutuhan	LKRB-1-07
Nama Kasus Uji	Menyimpan Testimoni
Prosedur	1. Telah berhasil login 2. Mengakses halaman Testimoni. 3. Aktor mengisi <i>form</i> testimoni yang terdiri dari nama, lokasi objek wisata, deskripsi dan foto.
Expected Result	Sistem menyimpan data testimoni pada basis data dan kemudian menampilkan data testimoni terbaru pada halaman testimoni.
Result	Sistem menyimpan data testimoni pada basis data dan kemudian menampilkan data testimoni terbaru pada halaman testimoni.
Status	Valid

Tabel 6.18 merupakan tabel untuk pengujian validasi *use case* menyimpan testimoni alternatif pertama.

Tabel 6.18 Pengujian Validasi Menyimpan Testimoni Alternatif 1

Kode Kebutuhan	LKRB-1-07
Nama Kasus Uji	Menyimpan Testimoni
Prosedur	1. Telah berhasil login 2. Mengakses halaman Testimoni. 3. Aktor tidak mengisi salah satu field atau seluruh field yang ada pada form testimoni
Expected Result	Sistem akan menampilkan pesan "please, fill out this field".

Result	Sistem akan menampilkan pesan “please, fill out this field”.
Status	Valid

Tabel 6.19 merupakan tabel untuk pengujian validasi *use case login* admin.

Tabel 6.19 Pengujian Validasi Login Admin

Kode Kebutuhan	LKRB-1-08
Nama Kasus Uji	<i>Login Admin</i>
Prosedur	Telah mengakses halaman <i>login</i> admin dan mengisi <i>form login</i> dengan mengisi <i>username</i> dan <i>password</i> , kemudian memilih tombol <i>login</i> .
Expected Result	Sistem menampilkan halaman utama admin.
Result	Sistem menampilkan halaman utama admin.
Status	Valid

Tabel 6.20 merupakan tabel untuk pengujian validasi *use case login* admin alternatif pertama.

Tabel 6.20 Pengujian Validasi Login Admin Alternatif 1

Kode Kebutuhan	LKRB-1-08
Nama Kasus Uji	<i>Login Admin</i>
Prosedur	Telah mengakses halaman <i>login</i> admin dan mengisi <i>form login</i> dengan <i>password</i> dan <i>username</i> tidak sesuai dengan yang tersimpan ada basis data.
Expected Result	Sistem menampilkan pesan “Error! Username atau password yang anda masukan salah”.
Result	Sistem menampilkan pesan “Error! Username atau password yang anda masukan salah”.
Status	Valid

Tabel 6.21 merupakan tabel untuk pengujian validasi *use case logout* admin.

Tabel 6.21 Pengujian Validasi Logout Admin

Kode Kebutuhan	LKRB-1-09
Nama Kasus Uji	<i>Logout Admin</i>
Prosedur	Memilih <i>logout</i> pada menu profil.
Expected Result	Sistem menampilkan halaman <i>login</i> admin.

Result	Sistem menampilkan halaman <i>login</i> admin.
Status	Valid

Tabel 6.22 merupakan tabel untuk pengujian validasi *use case* menambah objek wisata.

Tabel 6.22 Pengujian Validasi Menambah Objek Wisata

Kode Kebutuhan	LKRB-1-10
Nama Kasus Uji	Menambah Objek Wisata
Prosedur	1. Mengakses halaman tambah objek wisata 2. Mengisi <i>form</i> pada halaman tambah objek wisata yang terdiri dari nama lokasi, keterangan, lintang dan bujur dan kemudian memilih simpan.
Expected Result	Sistem menyimpan data pada basis data dan menampilkan data terbaru pada halaman daftar objek wisata.
Result	Sistem menyimpan data pada basis data dan menampilkan data terbaru pada halaman daftar objek wisata.
Status	Valid

Tabel 6.23 merupakan tabel untuk pengujian validasi *use case* menambah objek wisata alternatif pertama.

Tabel 6.23 Pengujian Validasi Menambah Objek Wisata Alternatif 1

Kode Kebutuhan	LKRB-1-10
Nama Kasus Uji	Menambah Objek Wisata
Prosedur	1. Mengakses halaman tambah objek wisata 2. Tidak mengisi salah satu <i>field</i> pada <i>form</i> tambah objek wisata yang terdiri dari nama lokasi, keterangan, lintang dan bujur dan kemudian memilih simpan.
Expected Result	Sistem menampilkan pesan "File Gambar Tidak Sesuai".
Result	Sistem menampilkan pesan "File Gambar Tidak Sesuai".
Status	Valid

Tabel 6.24 merupakan tabel untuk pengujian validasi *use case* menambah objek wisata alternatif kedua.

Tabel 6.24 Pengujian Validasi Menambah Objek Wisata Alternatif 2

Kode Kebutuhan	LKRB-1-10
Nama Kasus Uji	Menambah Objek Wisata
Prosedur	1. Mengakses halaman tambah objek wisata 2. Mengisi nama lokasi objek wisata sama dengan yang sudah terdaftar pada basis data.
Expected Result	Sistem menampilkan pesan “Nama Lokasi Telah Tersedia”.
Result	Sistem menampilkan pesan “Nama Lokasi Telah Tersedia”.
Status	Valid

Tabel 6.25 merupakan tabel untuk pengujian validasi *use case* mengubah detail informasi objek wisata.

Tabel 6.25 Pengujian Validasi Mengubah Detail Informasi Objek Wisata

Kode Kebutuhan	LKRB-1-11
Nama Kasus Uji	Mengubah Detail Informasi Objek Wisata
Prosedur	1. Mengakses halaman daftar objek wisata. 2. Memilih tombol yang berfungsi untuk melakukan perubahan data pada objek wisata yang ingin dilakukan perubahan data informasinya. 3. Mengisi <i>form</i> yang terdiri dari nama lokasi, keterangan, lintang dan bujur.
Expected Result	Sistem menyimpan data terbaru dan menampilkan data terbaru pada halaman daftar objek wisata.
Result	Sistem menyimpan data terbaru dan menampilkan data terbaru pada halaman daftar objek wisata.
Status	Valid

Tabel 6.26 merupakan tabel untuk pengujian validasi *use case* menghapus objek wisata.

Tabel 6.26 Pengujian Validasi Menghapus Objek Wisata

Kode Kebutuhan	LKRB-1-12
Nama Kasus Uji	Menghapus Objek Wisata
Prosedur	1. Mengakses halaman daftar objek wisata. 2. Memilih tombol yang berfungsi untuk melakukan hapus objek wisata. 3. Memilih “ok” pada pesan konfirmasi.

Expected Result	Sistem menghapus objek wisata yang telah di pilih dan menampilkan kembali daftar objek wisata dengan data terbaru.
Result	Sistem menghapus objek wisata yang ada di basis data dan menampilkan kembali daftar objek wisata dengan data terbaru.
Status	Valid

Tabel 6.27 merupakan tabel untuk pengujian validasi *use case* melihat daftar objek wisata.

Tabel 6.27 Pengujian Validasi Melihat Daftar Objek Wisata

Kode Kebutuhan	LKRB-1-13
Nama Kasus Uji	Melihat Daftar Objek Wisata.
Prosedur	Mengakses halaman daftar objek wisata.
Expected Result	Sistem menampilkan tabel daftar objek wisata pada halaman daftar objek wisata.
Result	Sistem menampilkan tabel daftar objek wisata pada halaman daftar objek wisata.
Status	<i>Valid.</i>

Tabel 6.28 merupakan tabel untuk pengujian validasi *use case* melihat grafik kepuasan pengunjung.

Tabel 6.28 Pengujian Validasi Melihat Grafik Kepuasan Pengunjung

Kode Kebutuhan	LKRB-1-14
Nama Kasus Uji	Melihat Grafik Kepuasan Pengunjung.
Prosedur	Mengakses halaman kelola objek wisata.
Expected Result	Sistem menampilkan grafik kepuasan pengunjung.
Result	Sistem menampilkan grafik kepuasan pengunjung.
Status	<i>Valid.</i>

6.4 Pengujian Kompatibilitas

Pengujian kompatibilitas dapat dilihat pada Tabel 6.29 dan hasil pengujian menggunakan *tools* SortSite dapat dilihat pada Gambar 6.7.

Tabel 6.29 Pengujian Kompatibilitas

Kode Kebutuhan	LKRB-2-001
Prosedur	Menggunakan sistem di beberapa browser yang berbeda.
Expected Result	Sistem dapat dijalankan pada 8 browser yang berbeda.
Result	Sistem dapat dijalankan pada 8 browser yang berbeda.
Status	<i>Valid</i>

Browser	Internet Explorer	Edge	Firefox	Safari	Opera	Chrome	iOS	Android	
Version	8 9 10 11	16	60	≤ 10 11	51	66	≤ 9 10 11	≤ 3 4*	
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Priority	Description and URL	Guideline and Line#	Count
Expand all 0 issues			

Gambar 6.7 Hasil Pengujian Kompatibilitas

Hasil dari pengujian kompatibilitas pada Gambar 6.7 dapat mengetahui apakah perangkat lunak yang dikembangkan memiliki masalah ketika dijalankan di 8 browser yang berbeda. Pada pengujian kompatibilitas ini terdapat beberapa permasalahan yang kemudian dikelompokkan menjadi tiga kategori yaitu:

1. *Critical Issues* digunakan untuk mengecek fungsionalitas dan kehilangan dari konten web.
2. *Major Issues* digunakan untuk mengecek layout utama seperti letak menu atau navigasi, ukuran gambar dan bentuk tabel.
3. *Minor Issues* digunakan untuk mengecek property pada layout yang ada di halaman seperti fungsi *required* pada *input* dan properti css atau tampilan.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut :

1. Berdasarkan hasil dari analisis kebutuhan dan analisis data, sistem Layanan Online KebunRaya Bogor memiliki 14 kebutuhan fungsional yang dapat membantu pihak PKT Kebun Raya-LIPI memberikan informasi dan pelayanan untuk pengunjung kebun raya Bogor, serta mempermudah dalam hal pengelolaan objek wisata. Kebutuhan fungsional yang didapat telah mencakup permasalahan yang ada yaitu menyediakan informasi pemetaan lokasi objek wisata, menyediakan informasi lokasi objek wisata terfavorit, menyediakan fitur untuk menyimpan testimoni dan menyediakan grafik kepuasan pengunjung kebun raya Bogor. Kebutuhan-kebutuhan tersebut nantinya dapat memberikan solusi untuk masalah dalam proses penggalan dan anilisi kebutuhan. Kemudian dari hasil analisis yang di dapatkan di modelkan dengan *Entity Relationship Diagram (ERD)*, *use case diagram* dan *use case scenario*.
2. Berdasarkan perancangan yang telah dilakukan, didapatkan hasil dari perancangan arsitektur adalah rancangan *sequence diagram* dan *class diagram*. Dalam perancangan komponen diperoleh rancangan algoritme-algoritme utama yang digunakan dalam sistem. Dalam perancangan data diperoleh rancangan basisdata berupa *Physical Data Model (PDM)*. Dan perancangan antarmuka berisi rancangan layout dari user interface sistem. Berdasarkan implementasi yang telah dilakukan, dihasilkan spesifikasi sistem, kode program, basis data, dan antarmuka. Pada spesifikasi sistem terdapat spesifikasi perangkat lunak, perangkat keras, sistem operasi, dan arsitektur sistem. Pada implementasi kode program dilakukan implementasi kelas-kelas dan algoritme-algoritme yang ada pada tahap perancangan. Pada implementasi data diperoleh query untuk mengimplementasikan perancangan database. Dan pada implementasi antarmuka diperoleh tampilan web yang digunakan untuk interaksi antara sistem dan pengguna sistem.
3. Berdasarkan dari hasil pengujian didapatkan, pengujian unit, pengujian integrasi dan pengujian validasi menghasilkan nilai 100% *valid*. Pada pengujian kompatibilitas di dapatkan hasil bahwa sistem dapat di berjalan pada 8 jenis perambah.

7.2 Saran

Saran yang diberikan untuk pengembangan lanjut perangkat lunak Layanan Online Kebun Raya Bogor ini selanjutnya antara lain :

1. Sistem dapat dikembangkan kembali dalam hal pendeteksian lokasi berdasarkan hasil dari foto yang di unggah oleh pengunjung.
2. Sistem dapat ditambahkan fitur obrolan online untuk pengunjung yang sedang mengakses website dengan admin.



DAFTAR PUSTAKA

- A.S Rosa dan Salahuddin M, 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Modula, Bandung.
- Amynarto, N., Sari, Y., & Wihandika, R. Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode Local Binary Pattern. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 10, p. 3230-3238, feb. 2018.
- Bradsky, Gary. 2008. Learning OpenCV. California: O'Reilly Media.
- Caytiles, Ronnie D. dan Lee, Sunguk., 2014. A Review of an MVC Framework based Software Development. International Journal of Software Engineering and Its Applications [SCOPUS]
- Connolly, Thomas and Begg, Carolyn. (2010). Database Systems A Practical Approach to Design, Implementation, and Management Fifth Edition. Boston: Pearson Education.
- Darwin, C. R. (1872). The Expression of The Emotions In Man and Animals. London: John Murray.
- Ekman, P. (1964). Body Position, Facial Expression, And Verbal Behavior During Interviews. Abnormal and Social Psychology, 295-301.
- Guru99, 2018. Learn McCabe's Cyclomatic Complexity with Example. [Online] Tersedia di: <<https://www.guru99.com/cyclomatic-complexity.html>> [Di akses 23 Maret 2018]
- Huang, X., Wang, S.-J., Zhao, G., & Pietikäinen, M. (2015). Facial Micro-Expression Recognition using Spatiotemporal Local Binary Pattern with Integral Projection. IEEE International Conference on Computer Vision Workshop, 1-9.
- Kelter, D., & Kring, A. (1998). Emotion, Social Function, and Psychopathology. Review of General Psychology, 320-342.
- Kindarto, Asdani. 2008. Asyik Berinternet dengan Beragam Layanan Google. Yogyakarta: Penerbit Andi.
- Kumar, Naresh et. al., 2013. Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction. International Journal of Soft Computing and Engineering (IJSCE), Vol. 3
- Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution Gray-scale and Rotation Invariant Texture Classification With Local Binary Patterns. IEEE Trans Pattern Anal Mach Intell 24, 971-987.
- Pagurawan, M.J, Rachmaniah,M., 2015. Pengembangan Sistem Informasi Tanaman di Kebun Raya Bogor Berbasis Android. Institut Pertanian Bogor.

- Pietikäinen, M., Hadid, A., Zhao, G., & Ahonen, T. (2011). Computer Vision Using Local Binary Pattern. 13-47.
- Pressman, Roger S., 2010. Software Engineering A Practitioner's Approach. 7th Ed. New York: McGraw-Hill.
- Republik Indonesia. 2011. Peraturan Presiden Nomor 93 Tahun 2011 tentang Kebun Raya (Lembaran Negara Republik Indonesia Tahun 2011 Nomor 143).
- Rumbaugh, James et al., 2005. The Unified Modeling Language Reference Manual. 2nd ed. Boston: Addison-Wesley.
- Sommerville, Ian., 2011. Software engineering. 9th ed. London: Addison-Wesley.
- Scherer, K. R. (1986). Vocal affect expression: A review and a model for future research. *Psychological Bulletin*, 99, 143- 165.
- Turiyanto, M. D., Purwanto, D., & Dikairono, R. (2014). Penerapan Teknik Pengenalan Wajah Berbasis Fitur Local Binary Pattern pada Robot Pengantar Makanan. 1-6.
- Yan, W.-J., Li, X., Wang, S.-J., Zhao, G., Liu, Y.-J., Chen, Y.-H., & Fu, X. (2014). CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation. 1-9.

